



Question: 15/16

STUDY GROUP 16 – CONTRIBUTION ____

SOURCE*: RAPPORTEUR FOR Q.15/16 (Gary SULLIVAN)

TITLE: DRAFT FOR “H.263++” ANNEXES U, V, AND W TO RECOMMENDATION H.263

This contribution contains a draft for three new annexes to be added to ITU-T Recommendation H.263 (*Video Coding for Low Bit Rate Communication*):

- Annex U**, specifying an optional *Enhanced Reference Picture Selection* (ERPS) mode,
- Annex V**, specifying an optional *Data Partitioned Slice* (DPS) mode, and
- Annex W**, specifying optional *Additional Supplemental Enhancement Information*.

Drafts of these annexes were determined by ITU-T SG16 in February, 2000. It is proposed that this draft of these annexes be decided at the SG 16 meeting in November 2000.

SUMMARY

This document contains three additional annexes to Recommendation H.263:

- Annex U**: An optional *Enhanced Reference Picture Selection* (ERPS) mode capable of providing enhanced coding efficiency and enhanced error resilience (particularly against loss of data packets). The ERPS mode operates by managing a multi-picture buffer of stored pictures.
- Annex V**: An optional *Data Partitioned Slice* (DPS) mode capable of providing enhanced error resilience (particularly against localized corruption of bitstream contents during transmission). The DPS mode operates by separating header and motion vector data from DCT coefficient data in the bitstream and by protecting motion vector data using a reversible representation.
- Annex W**: Optional *Additional Supplemental Enhancement Information* which can be added to an H.263 bitstream to provide backward-compatible enhancements, including:
 - Indication of use of a specific fixed-point IDCT
 - Picture Messages, including the message types of:
 - Arbitrary Binary Data,
 - Text (Arbitrary, Copyright, Caption, Video Description, or Uniform Resource Identifier),
 - Picture Header Repetition (Current, Previous, Next with Reliable Temporal Reference, or Next with Unreliable Temporal Reference),
 - Interlaced Field Indications (Top or Bottom), and
 - Spare Reference Picture Identification.

Annex U

Enhanced Reference Picture Selection mode

(This annex forms an integral part of this Recommendation.)

U.1 Introduction

This annex describes the optional Enhanced Reference Picture Selection (ERPS) mode of this Recommendation. The capability to use this optional mode is negotiated by external means (for example, Recommendation H.245). The amount of picture memory accommodated in the decoder for ERPS operation should also be signaled by external means. The use of this mode shall be indicated by setting the formerly-reserved bit 16 of the optional part of the PLUSPTYPE (OPPTYPE) to "1". The mode provides benefits for both error resilience and coding efficiency by using a memory buffer of reference pictures.

A sub-mode of the ERPS mode is specified for Sub-Picture Removal. The purpose of Sub-Picture Removal is to reduce the amount of memory required to store multiple reference pictures. The memory reduction is accomplished by specifying the partitioning of each reference picture into smaller rectangular units called sub-pictures. The encoder can then indicate to the decoder that specific sub-picture areas of specific reference pictures will not be used as a reference for the prediction of subsequent pictures, thus allowing the memory allocated in the decoder for storing these areas to be used to store data from other reference pictures. The support for this sub-mode and the allowed fragmentation of the picture memory into minimum picture units (MPUs) for Sub-Picture Removal as defined herein is also negotiated by external means (for example, Recommendation H.245).

A sub-mode of the ERPS mode is specified for enabling two-picture backward prediction in B pictures. This sub-mode can enhance performance by providing encoders for B pictures not only with an ability to use multiple references for forward prediction, but also to use more than one reference picture for backward prediction. The support for this sub-mode is negotiated by external means (for example, Recommendation H.245).

For error resilience, the ERPS mode can use backward channel messages, which are signaled by external means (for example, Recommendation H.245) sent from a decoder to an encoder to inform the encoder which pictures or parts of pictures have been incorrectly decoded. The ERPS mode provides enhanced performance compared to the Reference Picture Selection (RPS) mode defined in Annex N. It shall not be used simultaneously with the RPS mode. (It can be used in such a way as to provide essentially the same functionality as the RPS mode.)

For coding efficiency, motion compensation can be extended to prediction from multiple pictures. The extension of motion compensation to multi-picture prediction is achieved by extending each motion vector by a picture reference parameter that is used to address a macroblock or block prediction region for motion compensation in any of the multiple reference pictures. The picture reference parameter is a variable length code specifying a relative buffer index. The reference pictures are assembled in a buffering scheme that is controlled by the encoder.

The ERPS mode shall not be used with the Syntax-based Arithmetic Coding mode (see Annex E) or the Data Partitioned Slice mode (see Annex V).

Once activated, the ERPS mode shall not be inactivated in subsequent pictures in the bitstream unless the initial inactivation occurs in an I or EI picture and any reactivation is also in an I or EI picture and is accompanied by a buffer reset (RESET equal to "1"). If inactivated, the entire contents of the ERPS multi-picture buffer shall be set to "unused" status.

U.2 Video source coding algorithm

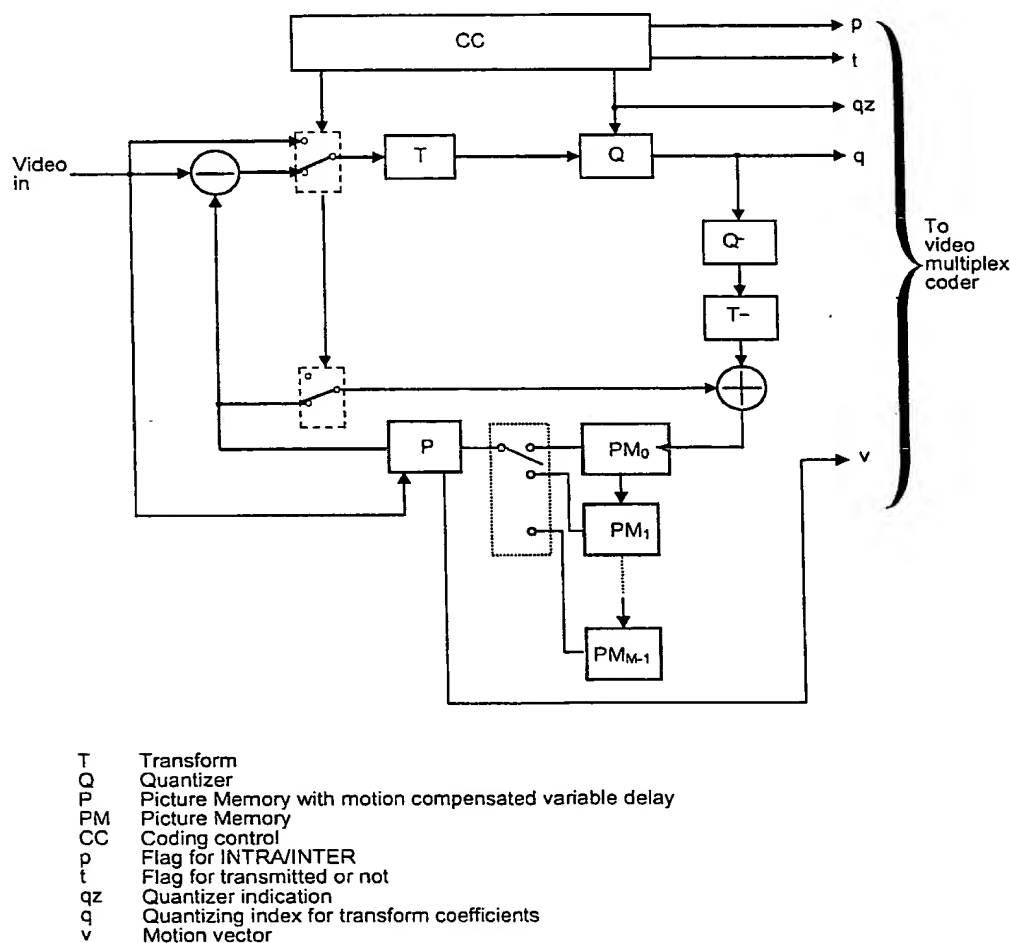
The source coder of this mode is shown in generalized form in Figure U.1. This figure shows a structure that uses a number of picture memories.

The video source coding algorithm can be extended to multi-picture motion compensation. Enhanced coding efficiency may be achieved by allowing reference picture selection on the macroblock level. A picture buffering scheme with relative indexing is employed for efficient addressing of pictures in the multi-picture buffer. The multi-picture buffer control may work in two distinct types of operation.

In the first of these two types of operation, a "Sliding Window" over time can be accommodated by the buffer control unit. In such a buffering scheme using M picture memories $PM_0 \dots PM_{M-1}$, the most recent preceding (up to M) decoded and reconstructed pictures are stored in the picture memories and can be used as references for decoding. If the number of pictures maximally accommodated by the multi-picture buffer corresponds to M , the motion estimation when coding a picture m , if $0 \leq m \leq M-1$, can utilize m pictures. When coding a picture $m \geq M$, the maximum number of pictures M can be used. Alternatively, a

second "Adaptive Memory Control" type of operation can be used for a more flexible and specific control of the picture memories than with the simple "Sliding Window" scheme.

The source coder may select one or several of the picture memories to suppress temporal error propagation caused by inter-picture coding. The Independent Segment Decoding mode (see Annex R), which treats boundaries of GOBs with non-empty headers or slices as picture boundaries, can be used to avoid spatial error propagation due to motion compensation across the boundaries of the GOBs or slices when this mode is applied to a smaller unit than a picture, such as a GOB or slice. The information to signal which picture is selected for prediction is included in the encoded bitstream.



Source coder for Enhanced Reference Picture Selection mode

Picture Selection layer (ERPS layer) is inserted. In the macroblock layer, picture reference parameters are inserted under certain conditions to enable multi-picture motion compensation.

U.3.1 Syntax of the Picture, GOB, and Slice layer

The Enhanced Reference Picture Selection syntax for the PLUS header (otherwise as shown in Figure 8) is shown in Figure U.2. The fields of RPSMF, PN, and the ERPS layer are inserted into the PLUS header. The fields of TRPI, TRP, BCI, and BCM are not present (since they are only needed for the RPS mode of Annex N, which is not allowed when the ERPS mode is active).

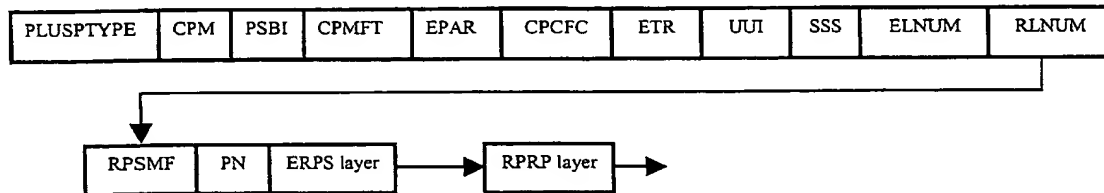


FIGURE U.2/H.263

Structure of PLUS Header for the ERPS mode

The syntax for the GOB layer is shown in Figure U.3. The fields of PNI, PN, NOERPSL, and the ERPS layer are added to the syntax (otherwise defined as in Figure 9).

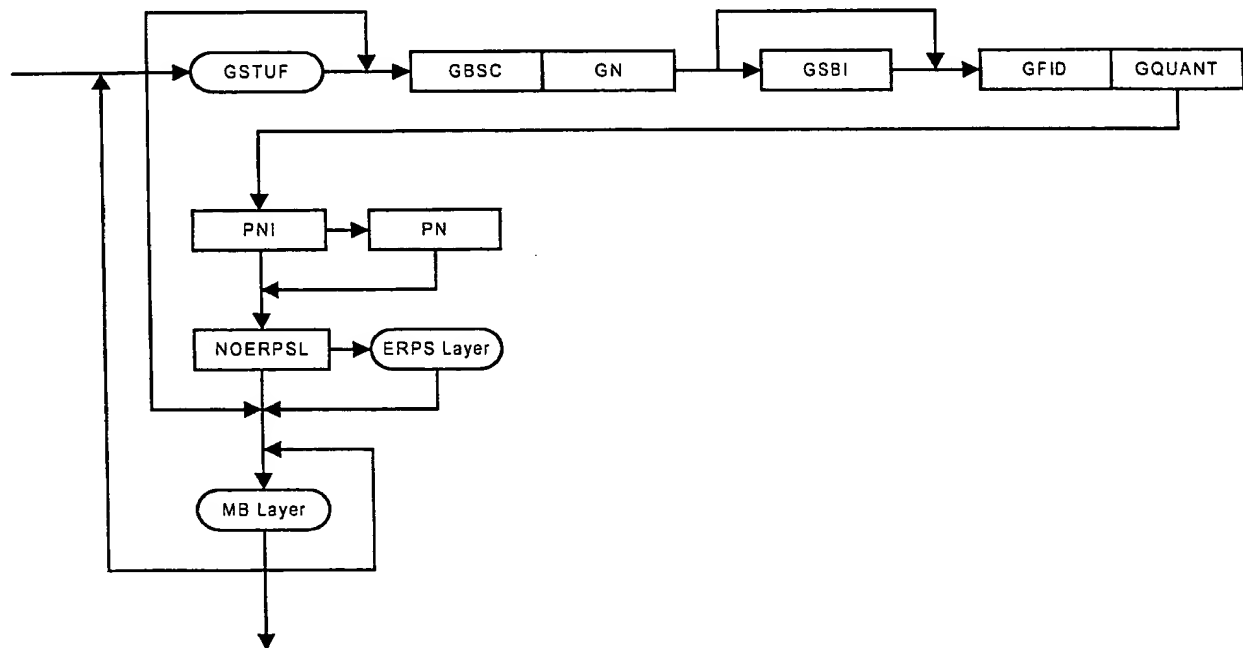


FIGURE U.3/H.263

Structure of GOB layer for the ERPS mode

When the optional Slice Structured mode (see Annex K) is in use, the syntax of the slice layer is modified in the same way as the GOB layer. The syntax for the slice layer is shown in Figure U.4. The slice that immediately follows the picture start code in the bitstream also includes all of the added fields PNI, PN, NOERPSL, and the ERPS layer.

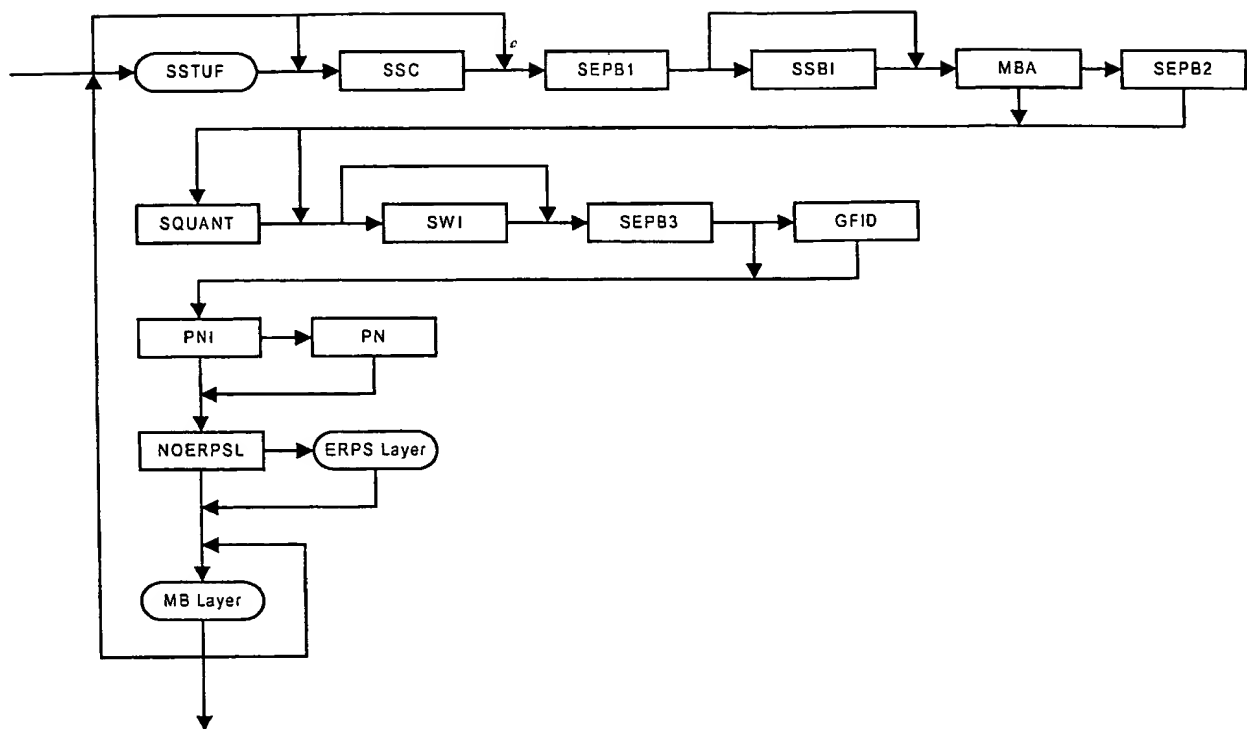


FIGURE U.4/H.263

Structure of Slice layer for the ERPS mode

The ERPS layer is shown in Figure U.5.

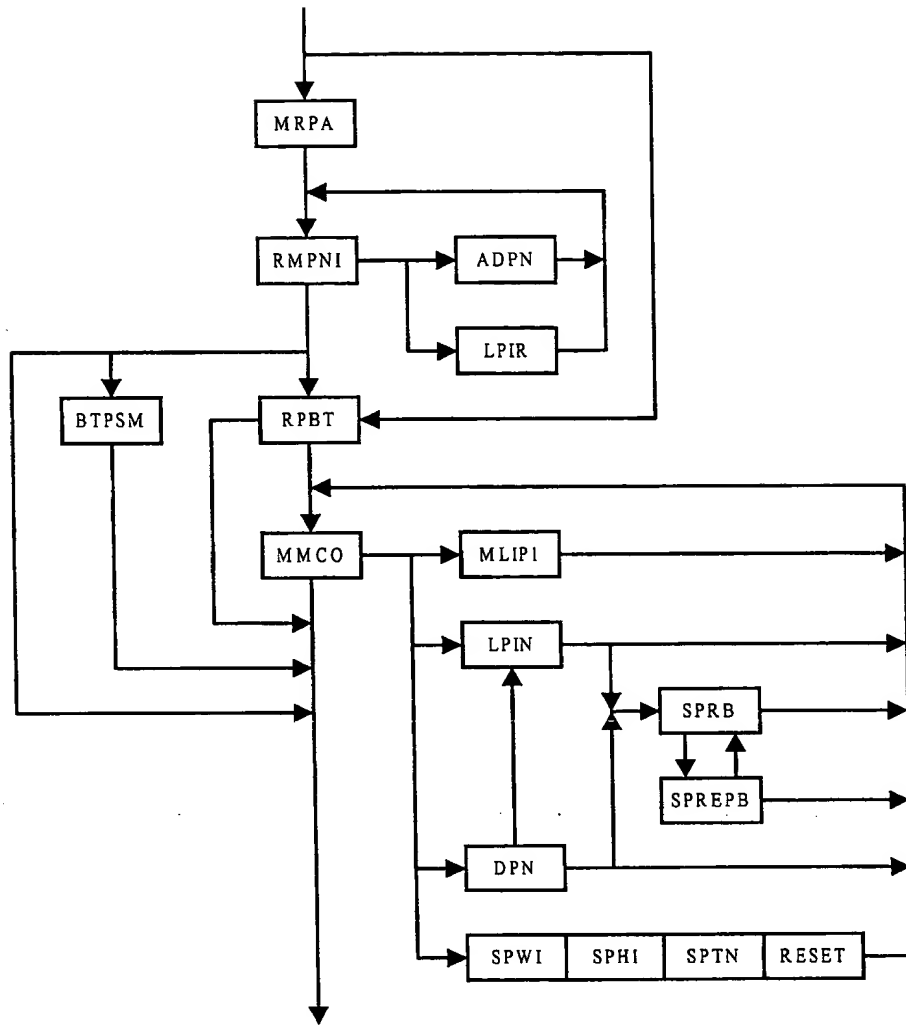


FIGURE U.5/H.263

Structure of the ERPS layer

Variable length codes for the ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR₀, PR₂, PR₃, PR₄, PRB, and PRFW fields are given in Table U.1.

Table U.1/H.263

Variable length codes for ADPN, LPIR, MLIP1, DPN, LPIN, SPTN, PR, PR₀, PR₂, PR₃, PR₄, PRB, and PRFW

| Absolute position | Number of bits | Codes |
|---|----------------|--|
| 0 | 1 | 1 |
| "x ₀ " + 1 (1:2) | 3 | 0x ₀ 0 |
| "x ₁ x ₀ " + 3 (3:6) | 5 | 0x ₁ 1x ₀ 0 |
| "x ₂ x ₁ x ₀ " + 7 (7:14) | 7 | 0x ₂ 1x ₁ 1x ₀ 0 |
| "x ₃ x ₂ x ₁ x ₀ " + 15 (15:30) | 9 | 0x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₄ x ₃ x ₂ x ₁ x ₀ " + 31 (31:62) | 11 | 0x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 63 (63:126) | 13 | 0x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 127 (127:254) | 15 | 0x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 255 (255:510) | 17 | 0x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 511 (511:1022) | 19 | 0x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₉ x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 1023 (1023:2046) | 21 | 0x ₉ 1x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |
| "x ₁₀ x ₉ x ₈ x ₇ x ₆ x ₅ x ₄ x ₃ x ₂ x ₁ x ₀ " + 2047 (2047:4094) | 23 | 0x ₁₀ 1x ₉ 1x ₈ 1x ₇ 1x ₆ 1x ₅ 1x ₄ 1x ₃ 1x ₂ 1x ₁ 1x ₀ 0 |

U.3.1.1 Reference Picture Selection Mode Flags (RPSMF) (3 bits)

RPSMF is a 3 bit fixed length codeword that is present in the PLUS header whenever the ERPS mode is in use (regardless of the value of UFEP). RPSMF shall not be present in the GOB or slice layer. When present, RPSMF indicates which type of back-channel messages are needed by the encoder. The values of RPSMF shall be as defined in subclause 5.1.13.

U.3.1.2 Picture Number Indicator (PNI) (1 bit)

PNI is a single bit fixed length codeword that is always present at the GOB or slice layer and is not present in the PLUS header. When present, PNI indicates whether or not the following PN field is also present.

"0": PN field is not present.

"1": PN field is present.

U.3.1.3 Picture Number (PN) (10 bits)

PN is a 10 bit fixed length codeword that is always present in the PLUS header when the ERPS mode is in use, and is present at the GOB or slice layer only when indicated by PNI.

PN shall be incremented by 1 for each coded and transmitted picture, in a 10-bit modulo operation, relative to the PN of the previous stored picture. The term "stored picture" is defined in subclause U.3.1.5.7. For EI and EP pictures, PN shall be incremented from the value in the last stored EI or EP picture within the same scalability enhancement layer. For B pictures, PN shall be incremented from the value in the most temporally-recent stored non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). B pictures are not stored in the multi-picture buffer, as they are not used as references for subsequent pictures. Thus a picture immediately following a B picture in the reference layer of the B picture or another B picture which immediately follows a B picture in the same enhancement layer shall have the same PN as the B picture. Similarly, if a non-B picture is present in the bitstream which is not stored, the picture following this non-B picture (in the same enhancement layer, in the case of Annex O operation) shall have the same PN as the non-stored non-B picture.

In a usage scenario known as "Video Redundancy Coding", the ERPS mode may be used by some encoders in a manner in which more than one representation is sent for the pictured scene at the same temporal instant (usually using different reference pictures). In such a case in which the ERPS mode is in use and in which adjacent pictures in the bitstream have the same temporal reference and the same picture number, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and shall decode and use the first such received picture while discarding the subsequent redundant picture(s).

The PN serves as a unique ID for each picture stored in the multi-picture buffer (for a given enhancement layer, in the case of Annex O operation) within 1024 coded and stored pictures. Therefore, a picture cannot be kept in the buffer after more than 1023 subsequent coded and stored pictures (in the same enhancement layer, in the case of Annex O operation) unless it has

been assigned a long-term picture index as specified below. The encoder shall ensure that the bitstream shall not specify retaining any short-term picture after more than 1023 subsequent stored pictures. A decoder which encounters a picture number on a current picture having a value equal to the picture number of some other short-term stored picture in the multi-picture buffer (in the same enhancement layer, in the case of Annex O operation, and excluding the video redundancy coding case described in the previous paragraph) should treat this condition as an error.

U.3.1.4 No Enhanced Reference Picture Selection Layer (NOERPSL) (1 bit)

NOERPSL is a single bit fixed length codeword that is present at the GOB or slice level whenever the ERPS mode is in use. It is not present in the PLUS header. The values of NOERPSL shall be as follows:

"0": The ERPS layer is sent,

"1": The ERPS layer is not sent.

If NOERPSL is "1", all ERPS settings and re-mappings in effect for the picture shall be applied also for the relevant GOB or slice. ERPS layer information sent at the GOB or slice level does not affect the decoding process of any other GOB or slice.

U.3.1.5 Enhanced Reference Picture Selection layer (ERPS) (variable length)

The ERPS layer is always present at the picture level when the ERPS mode is in use, and is present at the GOB or slice level if NOERPSL is "0". It specifies the buffer indexing used to decode the current picture, GOB, or slice, and manages the contents of the picture buffer.

U.3.1.5.1 Multiple Reference Pictures Active (MRPA) (1 bit)

MRPA is a single bit fixed length codeword that is present only if the picture coding type indicates a P-picture, an EP-picture, an Improved PB frame, or B-picture. MRPA is the first element in the ERPS layer if present. MRPA specifies whether the number of active reference pictures for forward-prediction or backward-prediction decoding of the current picture, GOB, or slice may be larger than one. The value of MRPA shall be as follows:

"1": More than one reference picture may be used for forward or backward motion compensation.

"0": Only one reference picture is used for forward or backward motion compensation. In this case, the extensions of the macroblock layer syntax in subclause U.3.2 do not apply.

MRPA may be changed from GOB to GOB or slice to slice so that different GOBs or slices may address different numbers of reference pictures.

MRPA shall be "0" in any picture which invokes the Reference Picture Resampling mode (see Annex P), and the same picture shall be indicated as the forward reference picture to be used at both the picture and GOB or slice levels for any such current picture. If the current picture is a B picture, the backward reference picture shall have the same size as the current picture, and any reference picture resampling process shall be applied only to the forward reference picture. Reference picture resampling shall be invoked only if the multi-picture buffer contains sufficient "unused" capacity to store the resampled forward reference picture, but after the resampled reference picture is used for the decoding of the current picture, the resampled forward reference picture shall not be stored in the multi-picture buffer.

U.3.1.5.2 Re-Mapping of Picture Numbers Indicator (RMPNI) (variable length)

RMPNI is a variable length codeword that is present in the ERPS layer if the picture is a P, EP, Improved PB, or B picture. RMPNI indicates whether any default picture indices are to be re-mapped for motion compensation of the current picture, GOB, or slice – and how the re-mapping of the relative indices into the multi-picture buffer is to be specified if indicated. RMPNI is transmitted using Table U.2. If RMPNI indicates the presence of an ADPN or LPIR field, an additional RMPNI field immediately follows the ADPN or LPIR field.

A picture reference parameter is a relative index into the ordered set of pictures. The RMPNI, ADPN, and LPIR fields allow the order of that relative indexing into the multi-picture buffer to be temporarily altered from the default index order for the decoding of a particular picture, GOB, or slice. The default index order is for the short-term pictures (i.e., pictures which have not been given a long-term index) to precede the long-term pictures in the reference indexing order. Within the set of short-term pictures, the default order is for the pictures to be ordered starting with the most recent buffered reference picture and proceeding through to the oldest reference picture (i.e., in decreasing order of picture number in the absence of wrapping of the ten-bit picture number field). Within the set of long-term pictures, the default order is for the pictures to be ordered starting with the picture with the smallest long-term index and proceeding up to the picture with long-term index equal to the most recent value of MLIP1-1.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order is:

- default relative index 0 refers to the short-term picture with picture number 303,
- default relative index 1 refers to the short-term picture with picture number 302,
- default relative index 2 refers to the short-term picture with picture number 300,
- default relative index 3 refers to the long-term picture with long-term picture index 0, and
- default relative index 4 refers to the long-term picture with long-term picture index 3.

The first ADPN or LPIR field that is received (if any) moves a specified picture out of the default order to the relative index of zero. The second such field moves a specified picture to the relative index of one, etc. The set of remaining pictures not moved to the front of the relative indexing order in this manner shall retain their default order amongst themselves and shall follow the pictures that have been moved to the front of the buffer in relative indexing order.

If MRPA is "0", no more than one ADPN or LPIR field shall be present in the same ERPS layer unless the current picture is a B picture. If the current picture is a B picture and MRPA is "0", no more than two ADPN or LPIR fields shall be present in the same ERPS layer.

Any re-mapping of picture numbers specified for some picture shall not affect the decoding process for any other picture. Any re-mapping of picture numbers specified for some GOB or slice shall not affect the decoding process for any other GOB or slice. A re-mapping of picture numbers specified for a picture shall only affect the decoding process for any GOB or slice within that picture in two ways:

- If NOERPSL is "1" at the GOB or slice level, then the re-mapping specified at the picture level is also used at the GOB or slice level.
- If the picture is a B picture, the re-mapping specified at the picture level shall specify the calculation of the value of TR_B and TR_D for direct bidirectional prediction.

An RMPNI "end loop" indication is the last element of the ERPS layer for a B picture if MRPA is "0". In a B picture with MRPA equal to "1", an RMPNI "end loop" indication is followed by BTPSM. In a P or EP picture or Improved PB frame, an RMPNI "end loop" indication is followed by RPBT.

Within one ERPS layer, RMPNI shall not specify the placement of any individual reference picture into more than one re-mapped position in relative index order.

Table U.2/H.263

RMPNI operations for re-mapping of reference pictures

| Value | Re-mapping Specified |
|-------|--|
| '1' | ADPN field is present and corresponds to a negative difference to add to a picture number prediction value |
| '010' | ADPN field is present and corresponds to a positive difference to add to a picture number prediction value |
| '011' | LPIR field is present and specifies the long-term index for a reference picture |
| '001' | End loop for re-mapping of picture relative indexing default order |

U.3.1.5.3 Absolute Difference of Picture Numbers (ADPN) (variable length)

ADPN is a variable length codeword that is present only if indicated by RMPNI. ADPN follows RMPNI when present. ADPN is transmitted using Table U.1, where the index into the table corresponds to $ADPN - 1$. ADPN represents the absolute difference between the picture number of the currently re-mapped picture and the prediction value for that picture number. If no previous ADPN fields have been sent within the current ERPS layer, the prediction value shall be the picture number of the current picture. If some previous ADPN field has been sent, the prediction value shall be the picture number of the last picture that was re-mapped using ADPN.

If the picture number prediction is denoted PNP, and the picture number in question is denoted PNQ, the decoder shall determine PNQ from PNP and ADPN in a manner mathematically equivalent to the following:

```
if (RMPNI == '1') { // a negative difference
  if (PNP - ADPN < 0)
    PNQ = PNP - ADPN + 1024;
  else
    PNQ = PNP - ADPN;
} else { // a positive difference
  if (PNP + ADPN > 1023)
    PNQ = PNP + ADPN - 1024;
  else
    PNQ = PNP + ADPN;
}
```

The encoder shall control RMPNI and ADPN such that the decoded value of ADPN shall not be greater than or equal to 1024.

As an example implementation, the encoder may use the following process to determine values of ADPN and RMPNI to specify a re-mapped picture number in question, PNQ:

```
DELTA = PNQ - PNP;
if (DELTA < 0) {
  if (DELTA < -511)
    MDELTA = DELTA + 1024;
  else
    MDELTA = DELTA;
} else {
  if (DELTA > 512)
    MDELTA = DELTA - 1024;
  else
    MDELTA = DELTA;
}
```

ADPN = abs(MDELTA);

where abs() indicates an absolute value operation. Note that the index into Table U.1 corresponds to the value of ADPN - 1, rather than the value of ADPN itself.

RMPNI would then be determined by the sign of MDELTA.

U.3.1.5.4 Long-term Picture Index for Re-Mapping (LPIR) (variable length)

LPIR is a variable length codeword that is present only if indicated by RMPNI. LPIR follows RMPNI when present. LPIR is transmitted using Table U.1. It represents the the long-term picture index to be re-mapped. The prediction value used by any subsequent ADPN re-mappings is not affected by LPIR.

U.3.1.5.5 B-Picture Two-Picture Prediction Sub-Mode (BTPSM) (1 bit)

BTPSM is a single bit fixed length codeword that is present only in a B picture (see Annex O) and only when MRPA is "1". It follows an RMPNI "end loop" indication and is the last element of the ERPS layer for the B picture when present. It indicates whether the two-picture backward prediction sub-mode is in use for the picture as follows:

- "0" : Single-picture backward prediction
- "1" : Two-picture backward prediction

BTPSM has an implied value of "0" if not present (when MRPA is "0").

The set of pictures available for use as forward prediction references is the set of pictures in the multi-picture buffer other than the set of backward reference pictures. The set of backward reference pictures is determined by the value of BTPSM. If single-picture backward prediction is specified by BTPSM, the first picture in (possibly re-mapped) relative index order is the only backward reference picture. If two-picture backward prediction is specified by BTPSM, the first two pictures in (possibly re-mapped) relative index order are the two backward reference pictures. The relative index for forward prediction then becomes a relative index into the set of forward reference pictures.

The contents of the multi-picture buffer are not affected by the presence of a B picture. The B picture is not stored in the multi-picture buffer and is not used as a reference for the coding of subsequent pictures.

U.3.1.5.6 Reference Picture Buffering Type (RPBT) (1 bit)

RPBT is a single bit fixed length codeword that specifies the buffering type of the currently decoded picture. It follows an RMPNI “end loop” indication when the picture is not an I, EI, or B picture. It is the first element of the ERPS layer if the picture is an I or EI picture. It is not present if the picture is a B picture. The values for RPBT are defined as follows:

“1” : Sliding Window,

“0” : Adaptive Memory Control.

In the “Sliding Window” buffering type, the current decoded picture shall be added to the buffer with default relative index 0, and any marking of pictures as “unused” in the buffer is performed automatically in a first-in-first-out fashion among the set of short-term pictures. In this case, if the buffer has sufficient “unused” capacity to store the current picture, no additional pictures shall be marked as “unused” in the buffer. If the buffer does not have sufficient “unused” capacity to store the current picture, the picture (or pictures as necessary to free the needed amount of memory in the case of sub-picture removal) with the largest default relative index (or indices as necessary in the case of sub-picture removal) among the short-term pictures in the buffer shall be marked as “unused”. In the sliding window buffering type, no additional information is transmitted to control the buffer contents.

In the “Adaptive Memory Control” buffering type, the encoder explicitly specifies any addition to the buffer or marking of data as “unused” in the buffer, and may also assign long-term indices to short-term pictures. The current picture and other pictures may be explicitly marked as “unused” in the buffer, as specified by the encoder. This buffering type requires further information that is controlled by memory management control operation (MMCO) parameters.

RPBT, if present in GOB or slice layers, shall be the same as in the picture layer. Any MMCO command present in GOB or slice layers shall convey the same operation as some MMCO command in the picture layer.

If the picture is a B picture, RPBT shall not be present and the decoded picture shall not be stored in the multi-picture buffer. This ensures that a B picture shall not affect the contents of the multi-picture buffer.

Similarly, the B-picture part of an Improved PB frame shall not be stored in the buffer. All control fields associated with controlling the storage of an Improved PB frame shall be considered to be associated with controlling the storage of only the P-picture part of the Improved PB frame.

U.3.1.5.7 Memory Management Control Operation (MMCO) (variable length)

MMCO is a variable length codeword that is present only when RPBT indicates “Adaptive Memory Control”, and may occur multiple times if present. It specifies a control operation to be applied to manage the multi-picture buffer memory. The MMCO parameter is followed by data necessary for the operation specified by the value of MMCO, and then an additional MMCO parameter follows – until the MMCO value indicates the end of the list of such operations. MMCO commands do not affect the buffer contents or the decoding process for the decoding of the current picture – rather, they specify the necessary buffer status for the decoding of subsequent pictures in the bitstream. The values and control operations associated with MMCO are defined in Table U.3.

All memory management control operations specified using MMCO shall be specified in the picture layer. Some or all of the same operations as are specified at the picture layer may also be specified at the GOB or slice layer (with the same associated data). MMCO shall not specify memory operations at the GOB or slice layer that are not also specified with the same associated data at the picture layer.

A buffer size and structure specification MMCO command shall be the first MMCO command if present. No more than one buffer size and structure specification MMCO command shall be present in a given ERPS layer. A buffer size and structure specification MMCO command with RESET equal to “1” shall be present in the first picture in which the ERPS mode is activated in any series of ERPS mode pictures in the bitstream. A buffer size and structure specification MMCO command with RESET equal to “1” shall precede any use of MMCO to indicate marking sub-picture areas of any short-term or long-term pictures as “unused”. The sub-picture width and height specified in a buffer size and structure specification MMCO command shall not differ from the value of these parameters in a prior buffer size and structure specification MMCO command unless the current picture is an I or EI picture with RESET equal to “1”. The picture height and width shall not change within the bitstream except within a picture containing a buffer size and structure specification MMCO command with RESET equal to “1” (or within a picture in which the ERPS mode is not in use).

If a B picture using single-picture backward prediction is present in the bitstream, exactly one temporally subsequent non-B picture in the reference layer of the B picture shall precede the B picture in bitstream order, as specified in subclause O.2. No memory management control operations shall be present within any ERPS layer of this immediately temporally subsequent

non-B picture within the reference layer of the B picture which mark any part of that immediately temporally succeeding non-B picture as “unused”, since that reference layer picture is needed for display until after the decoding of the B picture.

The transmission order constraints specified in subclause O.2 are adjusted as necessary for B pictures using two-picture backward prediction. If a B picture using two-picture backward prediction is present in the bitstream, exactly two temporally subsequent non-B pictures in the reference layer of the B picture shall precede the B picture in bitstream order. The other restrictions on the transmission order of the B picture in the bitstream specified in O.2 shall apply, but as adjusted for the use of two temporally subsequent reference layer pictures. No memory management control operations shall be present within any ERPS layer of these two immediately temporally subsequent two non-B pictures within the reference layer of the B picture which mark any part of these two non-B pictures as “unused”, since these reference layer pictures are needed for display until after the decoding of the B picture.

A “stored picture” is defined as a non-B picture which does not contain an MMCO command in its ERPS layer which marks that (entire) picture as “unused”. If the current picture is not a stored picture, its ERPS layer shall not contain any of the following types of MMCO commands:

- An MMCO command to specify the buffer size and structure with RESET equal to “1”,
- Any MMCO command which marks any other picture (other than the current picture) as “unused” that has not also been marked as “unused” in the ERPS layer of a prior stored picture,
- Any MMCO command which assigns a long-term index to a picture that has not also been assigned the same long-term index in the ERPS layer of a prior stored picture, or
- Any MMCO command which marks sub-picture areas of any picture as “unused” that have not also been marked as “unused” in the ERPS layer of a prior stored picture.

Table U.3/H.263

Memory Management Control Operation (MMCO) Values

| Value | Memory Management Control Operation | Associated Data Fields Following |
|---------|---|----------------------------------|
| '1' | End MMCO Loop | None (end of ERPS layer) |
| '011' | Mark a Short-Term Picture as “Unused” | DPN |
| '0100' | Mark a Long-Term Picture as “Unused” | LPIN |
| '0101' | Assign a Long-Term Index to a Picture | DPN and LPIN |
| '00100' | Mark Short-Term Sub-Picture Areas as “Unused” | DPN and SPRB |
| '00101' | Mark Long-Term Sub-Picture Areas As “Unused” | LPIN and SPRB |
| '00110' | Specify the Maximum Long-Term Picture Index | MLIP1 |
| '00111' | Specify the Buffer Size and Structure | SPWI, SPHI, SPTN, and RESET |

U.3.1.5.8 Difference of Picture Numbers (DPN) (variable length)

DPN is present when indicated by MMCO. DPN follows MMCO if present. DPN is transmitted using codewords in Table U.1 and is used to calculate the PN of a picture for a memory control operation. It is used in order to assign a long-term index to a picture, mark a short-term picture as “unused”, or mark sub-picture areas of a short-term picture as “unused”. If the current decoded picture number is PNC and the decoded value from Table U.1 is DPN, an operation mathematically equivalent to the following equations shall be used for calculation of PNQ, the specified picture number in question:

```

if (PNC – DPN < 0)
    PNQ = PNC – DPN + 1024;
else
    PNQ = PNC – DPN;

```

Similarly, the encoder may compute the DPN value to encode using the following relation:

```

if (PNC – PNQ < 0)
    DPN = PNC – PNQ + 1024;
else
    DPN = PNC – PNQ;

```

For example, if the decoded value of DPN is zero and MMCO indicates marking a short-term picture as “unused”, the current decoded picture shall be marked as “unused” (thus indicating that the current picture is not a stored picture).

U.3.1.5.9 Long-term Picture Index (LPIN) (variable length)

LPIN is present when indicated by MMCO. LPIN is transmitted using codewords in Table U.1 and specifies the long-term picture index of a picture. It follows DPN if the operation is to assign a long-term index to a picture. It follows MMCO if the operation is to mark a long-term picture as “unused” or to mark sub-picture areas of a long-term picture as “unused”.

U.3.1.5.10 Sub-Picture Removal Bit-Map (SPRB) (fixed length)

SPRB is a fixed length codeword that contains one bit for each sub-picture area of a picture and is present when indicated by MMCO. The number of bits of SPRB data is determined by the most recent values of SPWI and SPHI. SPRB is used to indicate which sub-picture areas of a buffered picture are to be marked as “unused”. SPRB follows DPN if the operation is to mark sub-picture areas of a short-term picture as “unused”, and follows LPIN if the operation is to mark sub-picture areas of a long-term picture as “unused”.

Sub-pictures are numbered in raster scan order starting from the upper-left corner of the picture. For example, consider a case in which a reference picture, specified by DPN, is partitioned into six sub-pictures. Let “ $s_1 s_2 s_3 s_4 s_5 s_6$ ” represent six bits of SPRB data. If bit s_i is “1”, then the decoder should mark the i^{th} sub-picture in the indicated reference picture as “unused”. For example, if the SPRB is ‘000110’, then the fourth and fifth sub-pictures areas are marked as “unused”.

To prevent start code emulation, all necessary SPREPB emulation prevention bits shall be inserted within or following the SPRB data as specified in subclause U.3.1.5.11.

If SPRB is present and the specified picture has previously been affected by a prior SPRB bit-map, the bit-map specified by SPRB shall contain a “1” for any sub-picture area that contained a “1” in that previous SPRB bit-map. Every SPRB bit-map shall contain at least one bit having the value “0” and at least one bit having the value “1”.

U.3.1.5.11 Sub-Picture Removal Emulation Prevention Bit (SPREPB) (one bit)

SPREPB is a single bit fixed length codeword having the value “1” which shall be inserted immediately after any string of 8 consecutive zero bits of SPRB data.

U.3.1.5.12 Maximum Long-Term Picture Index Plus 1 (MLIP1) (variable length)

MLIP1 is a variable length codeword that is present if indicated by MMCO. MLIP1 follows MMCO if present. MLIP1 is transmitted using codewords in Table U.1. If present, MLIP1 is used to determine the maximum index allowed for long-term reference pictures (until receipt of another value of MLIP1). The decoder shall initially assume MLIP1 is “0” until some other value has been received. Upon receiving an MLIP1 parameter, the decoder shall consider all long-term pictures having indices greater than the decoded value of MLIP1 – 1 as “unused” for referencing by the decoding process for subsequent pictures. For all other pictures in the multi-picture buffer, no change of status shall be indicated by MLIP1.

U.3.1.5.13 Sub-Picture Width Indication (SPWI) (7 bits)

SPWI is a fixed length codeword of 7 bits that is present if indicated by MMCO. SPWI follows MMCO when indicated. SPWI specifies the width of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture width is $16 \cdot (\text{SPWI} + 1)$ luminance samples. The current picture has a width in sub-picture units of $\text{ceil}(\text{ceil}(\text{pw} / 16) / (\text{SPWI} + 1))$ sub-pictures, where pw is the width of the picture and “/” indicates floating-point division. For positive numbers, the ceiling function, $\text{ceil}(x)$, equals x if x is an integer and otherwise $\text{ceil}(x)$ equals one plus the integer part of x . If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, Recommendation H.245), the sub-picture width specified by SPWI shall be an integer multiple of the width of the MPU; otherwise, the sub-picture width specified by SPWI shall be such that SPWI is equal to $\text{ceil}(\text{pw} / 16) - 1$.

U.3.1.5.14 Sub-Picture Height Indication (SPHI) (7 bits)

SPHI is a fixed length codeword of 7 bits that is present if SPWI is present (as indicated by MMCO). SPHI follows SPWI if present. SPHI specifies the height of a sub-picture in units of 16 luminance samples, such that the indicated sub-picture height is $16 \cdot \text{SPHI}$. The allowed range of values of SPHI is from 1 to 72. The current picture has a height of $\text{ceil}(\text{ceil}(\text{ph} / 16) / \text{SPHI})$ sub-pictures, where ph is the height of the picture and “/” indicates floating-point division. If a minimum picture unit (MPU) size defining the minimum width and height of a sub-picture has been negotiated by external means (for example, Recommendation H.245), the sub-picture height specified by SPHI shall be an integer multiple of the height of the MPU; otherwise, the sub-picture height specified by SPHI shall be such that SPHI is equal to $\text{ceil}(\text{ph} / 16)$.

U.3.1.5.15 Sub-Picture Total Number (SPTN) (variable length)

SPTN is a variable length codeword that is present if SPWI and SPHI are present (as indicated by MMCO). SPTN follows SPHI if present. SPTN is coded using Table U.1, where the index into Table U.1 corresponds to the decoded value of SPTN – 1. The decoded value of SPTN is the total operational size capacity of the multi-picture buffer in units of sub-pictures as

specified by SPWI and SPHI. The memory capacity needed for the decoding of current pictures is not included in SPTN – only the memory capacity needed for storing the reference pictures to use for the prediction of other pictures. When sub-picture removal is not in use (i.e. when SPWI and SPHI have whole-picture dimensions), the maximum number of active short-term reference pictures (for example, for sliding window operation) is thus given by SPTN minus the number of pictures that have been assigned to long-term indices and have not been subsequently marked as “unused”.

U.3.1.5.16 Buffer Reset Indicator (RESET) (1 bit)

RESET is a single bit fixed length codeword that is present if SPWI, SPHI, and SPTN are present (as indicated by MMCO). RESET follows SPTN if present. The values of RESET shall be as follows:

- "0": The buffer contents are not reset,
- "1": The buffer contents are reset.

If RESET is "1", all pictures in the multi-picture buffer (but not the current picture unless specified separately) shall be marked “unused” (including both short-term and long-term pictures).

U.3.2 Macroblock layer syntax

U.3.2.1 P-Picture and Improved PB frames Macroblock Syntax

The macroblock layer syntax is modified if the ERPS layer is present for P-pictures and Improved PB frames when the number of selected forward reference pictures may be greater than one, as indicated by MRPA. The field MRPA is signaled in the ERPS layer. The macroblock layer syntax is shown in Figure U.7 when MRPA is "1". Otherwise, the macroblock syntax format in a P picture or Improved PB frame is not altered from that shown in Figure 10.

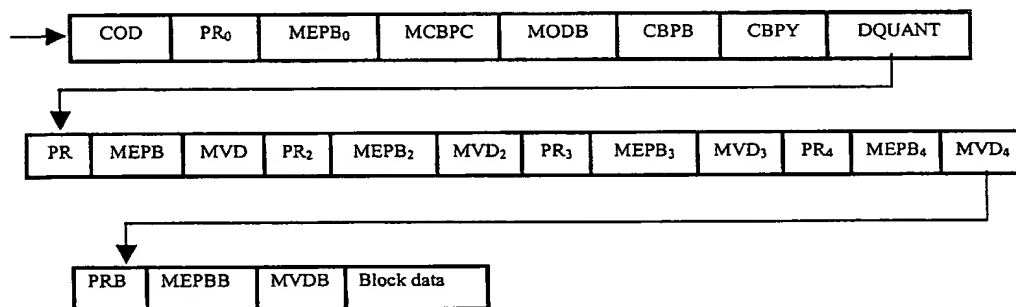


FIGURE U.7/H.263

Structure of P-picture and Improved PB frame Macroblock layer for the ERPS mode

U.3.2.1.1 Interpretation of COD

If the COD bit is "1", no further information is transmitted for the macroblock. In that case, the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the entire macroblock equal to zero, picture reference parameter equal to zero, and with no coefficient data. If the COD bit is "0", indicating that the macroblock is coded, the syntax of the macroblock layer is depicted in Figure U.7 with the fields PR₀, PR, PR₂, PR₃, PR₄, and PRB being included in the syntax. PR₀, PR, PR₂, PR₃, PR₄, and PRB each consist of a variable length codeword as given in Table U.1.

U.3.2.1.2 Picture Reference Parameter 0 (PR₀) (variable length)

PR₀ is a variable length codeword as specified in Table U.1. It is present whenever COD is "0". If PR₀ has a decoded value of zero (codeword "1"), it indicates that further information will follow for the macroblock. If decoded as non-zero, it indicates the coding of the macroblock using only a picture reference parameter.

If the field PR₀ does not have a decoded value of zero (codeword '1'), no further information is transmitted for this macroblock. In that case the decoder shall treat the macroblock as an INTER macroblock with the motion vector for the whole block equal to zero, the picture reference parameter equal to PR₀, and with no coefficient data.

If the field PR_0 has a decoded value of zero (codeword '1'), the macroblock is coded. The meaning and usage of the fields MCBPC, CBPB, CBPY, and DQUANT remains unaltered. The field PR is included together with the field MVD for all INTER macroblocks (and in Improved PB frames mode also for INTRA macroblocks). The use of MODB in Improved PB frames is described in subclause U.3.2.1.4.

U.3.2.1.3 Macroblock Emulation Prevention Bit 0 (MEPB₀) (1 bit)

MEPB₀ is a single bit fixed length codeword having the value "1" that follows PR_0 if and only if PR_0 is present and has a decoded value of "1" (codeword '000'), and either of the following two conditions are satisfied:

1. the slice structured mode (see Annex K) is in use, or
2. the COD for the current macroblock immediately follows after another macroblock which also has COD = "0" and PR_0 = "1" (codeword '000'), and the PR_0 of the previous macroblock is not followed by an MEPB₀ bit.

The purpose of MEPB₀ is to prevent start-code emulation and, in the slice structured mode, to aid in determining the number of macroblocks in a slice.

U.3.2.1.4 Macroblock Picture Reference Parameters (PR, PR₂₋₄, and PRB) (variable length)

PR is the primary picture reference parameter. PR is present whenever MVD is present. The three codewords PR₂₋₄ are included together with MVD₂₋₄ if indicated by PTYPE and if MCBPC specifies an INTER4V or INTER4V+Q macroblock (a macroblock of type 2 or 5 in Tables 8 and 9). PR₂₋₄ and MVD₂₋₄ are only present when in Advanced Prediction mode (see Annex F) or Deblocking Filter mode (see Annex J). PRB is only present in an Improved PB frame when MODB indicates that MVDB is present. PR, PR₂₋₄, and PRB each specify a picture reference relative index into the multi-picture buffer.

PR is used as the picture reference parameter for motion compensation of the entire macroblock if the macroblock is not an INTER4V or INTER4V+Q macroblock. If the macroblock is an INTER4V or INTER4V+Q macroblock, PR is used for motion compensated prediction of the first of the four 8×8 luminance blocks in the macroblock and for the two chrominance blocks of the macroblock (with the motion compensation process otherwise as specified in subclause 6.1). PR₂₋₄ are used for motion compensation of the remaining three 8×8 blocks of luminance data in the macroblock. If MODB indicates that MVDB is present, PRB is the picture reference parameter for forward prediction of the B part of the Improved PB frame.

In Improved PB frames when MODB indicates B_{PB} bidirectional prediction, the values of TR_D and TR_B shall be computed as the temporal reference increments based on the temporal reference data of the current picture and that of the most recent previous reference picture, regardless of whether or not the most recent previous reference picture has been re-mapped to a difference relative index order, marked as "unused", or assigned to a long-term index. The picture used as the forward reference picture for B_{PB} bidirectional prediction in Improved PB frames shall be the picture specified by PR.

U.3.2.1.5 Macroblock Emulation Prevention Bits (MEPB, MEPB₂₋₄, and MEPBB) (1 bit each)

MEPB, MEPB₂₋₄, and MEPBB are each a single bit having the value "1" if present. Each shall be present if and only if the Unrestricted Motion Vector mode (see Annex D) is not in use and the associated PR, PR₂₋₄, or PRB field is present and has the decoded value "1" (codeword '000'). Their purpose is to prevent start-code emulation.

U.3.2.2 B-Picture and EP-Picture Macroblock Syntax

The macroblock layer syntax for B and EP pictures (see Annex O) is modified in a similar fashion as in P pictures. The COD bit, if equal to "1", indicates a skipped macroblock as defined in Annex O, using a picture reference parameter of zero for the forward (skipped) prediction in an EP picture and for the forward part of direct (skipped) bidirectional prediction in a B picture and using the first backward prediction picture for the backward part of direct (skipped) bidirectional prediction in a B picture (in the case of two-picture backward prediction, as when BSBBW is present and equal to "0"). If COD is "0", a PR_0 parameter is inserted into the syntax and is used in a similar manner as described in U.3.2.1.2. If PR_0 is present and does not have a decoded value of zero (codeword '1'), it indicates that the macroblock is to be predicted with forward INTER prediction using a zero-valued motion vector and a picture reference parameter of PR_0 . If PR_0 has a decoded value of zero, MBTYPE follows and specifies the macroblock type. The format of the CBPC, CBPY, and DQUANT fields is unchanged. The MVDFW and MVDBW fields are encoded in the same manner as when the ERPS mode is not in use, but are each used in conjunction with a picture reference, and possibly an emulation prevention bit.

For a B picture, the backward reference pictures in the multi-picture buffer are defined as follows:

- In the case of single-picture backward prediction, there is only one backward reference picture, which is the first picture in (possibly re-mapped) relative index order, and
- In the case of two-picture backward prediction, there are two backward reference pictures, which are the first two pictures in (possibly re-mapped) relative index order.

The forward reference pictures in the multi-picture buffer are defined as the pictures in the multi-picture buffer other than the backward reference pictures. The relative indexing for forward prediction is a relative index into the forward reference picture set, and the relative indexing for backward prediction is a relative index into the backward reference picture set.

For example, if the buffer contains three short-term pictures with short-term picture numbers 300, 302, and 303 (which were transmitted in increasing picture-number order) and two long-term pictures with long-term picture indices 0 and 3, the default index order in the case of two-picture backward prediction is:

- default backward relative index 0 refers to the short-term picture with picture number 303,
- default backward relative index 1 refers to the short-term picture with picture number 302,
- default forward relative index 0 refers to the short-term picture with picture number 300,
- default forward relative index 1 refers to the long-term picture with long-term picture index 0, and
- default forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single default backward reference picture is the short-term picture with picture number 303,
- default forward relative index 0 refers to the short-term picture with picture number 302,
- default forward relative index 1 refers to the short-term picture with picture number 300,
- default forward relative index 2 refers to the long-term picture with long-term picture index 0, and
- default forward relative index 3 refers to the long-term picture with long-term picture index 3;

and if these pictures have been re-mapped to a new relative indexing order of short-term picture 302, followed by short-term picture 303, followed by long-term picture 0, followed by short-term picture 300, followed by long-term picture 3, the new relative index order in the case of two-picture backward prediction is:

- re-mapped backward relative index 0 refers to the short-term picture with picture number 302,
- re-mapped backward relative index 1 refers to the short-term picture with picture number 303,
- re-mapped forward relative index 0 refers to the long-term picture with long-term picture index 0,
- re-mapped forward relative index 1 refers to the short-term picture with picture number 300, and
- re-mapped forward relative index 2 refers to the long-term picture with long-term picture index 3;

and in the case of single-picture backward prediction:

- the single re-mapped backward reference picture is the short-term picture with picture number 302,
- re-mapped forward relative index 0 refers to the short-term picture with picture number 303,
- re-mapped forward relative index 1 refers to the long-term picture with long-term picture index 0,
- re-mapped forward relative index 2 refers to the short-term picture with picture number 300, and
- re-mapped forward relative index 3 refers to the long-term picture with long-term picture index 3.

The TR_D used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the first forward reference picture in (possibly re-mapped) relative index order and the first backward reference picture in (possibly re-mapped) relative index order (i.e. if two-picture backward prediction is in use, this would be the picture referenced when BSBBW is "0" as described in sub-clause U.3.2.2.3). The TR_B used for direct bidirectional prediction in a B picture shall be computed as the temporal reference increment between the B picture and the first forward reference picture in (possibly re-mapped) relative index order. The relative index order used in the computation of TR_D and TR_B shall be that specified by the ERPS layer at the picture level of the B picture syntax (i.e. re-mappings at the GOB or slice level shall not affect the values of TR_D and TR_B).

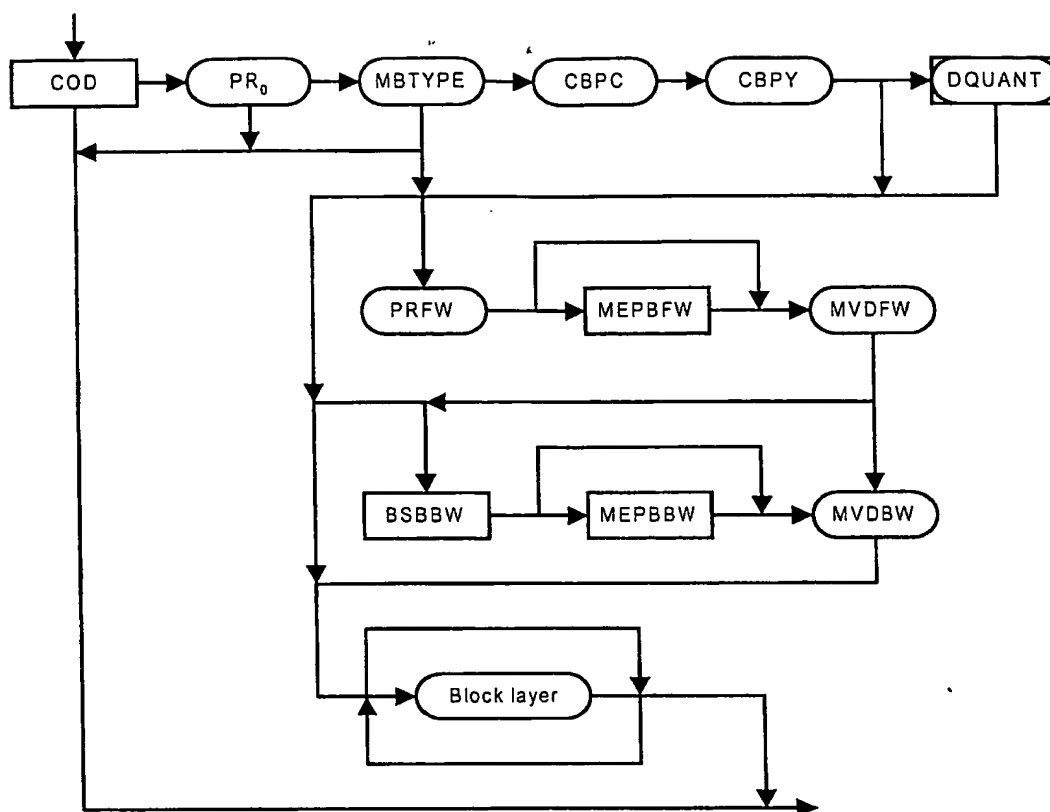


FIGURE U.8/H.263

Structure of EP and B picture Macrobloc layer for the ERPS mode

U.3.2.2.1 Picture Reference for Forward Prediction (PRFW) (variable length)

PRFW is a variable length picture reference parameter that is present whenever MVDFW is present, and is encoded using Table U.1. PRFW is a relative index into the set of forward reference pictures.

U.3.2.2.2 Emulation Prevention Bit for Forward Prediction (MEPBFW) (1 bit)

MEPBFW is a single bit fixed length codeword having the value "1" which shall be inserted after PRFW if and only if PRFW is present and has a decoded value of "1" (codeword '000') and the unrestricted motion vector mode (see Annex D) is not in use.

U.3.2.2.3 B-Picture Selection Bit for Backward Prediction (BSBBW) (1 bit)

BSBBW is a single bit fixed length codeword that is present only for B pictures when MVDBW is present and only when two-picture backward prediction is specified for the B picture operation. The meaning of this bit shall be defined as:

- "0" : Prediction from the first backward reference picture in relative index order (in default order, this would be the most recent short-term reference picture if that picture has not been assigned a long-term index or marked as "unused")
- "1" : Prediction from the second backward reference picture in relative index order (in default order, this would be the second-most recent short-term reference picture if neither of the last two reference pictures has been assigned a long-term index or marked as "unused")

U.3.2.2.4 Emulation Prevention Bit for Backward Prediction (MEPBW) (1 bit)

MEPBW is a single bit fixed length codeword having the value "1" that is present only under the following conditions:

- BSBBW is present and equal to "0", and
- The unrestricted motion vector mode (see Annex D) is not in use, and
- BSBBW is preceded by five bits having the value '00000'

U.4 Decoder Process

The decoder for the ERPS mode stores the reference pictures for inter-picture decoding in a multi-picture buffer. The decoder may need additional memory capacity to store the multiple decoded pictures (relative to the memory capacity needed without support of the ERPS mode). The decoder replicates the multi-picture buffer of the encoder according to the reference picture buffering type and any memory management control operations specified in the bitstream. The buffering scheme may also be operated when partially erroneous pictures are decoded.

Each transmitted and stored picture is assigned a Picture Number (PN) which is stored with the picture in the multi-picture buffer. PN represents a sequential picture counting identifier for stored pictures. PN is constrained, using modulo 1024 arithmetic operation. For the first transmitted picture, PN should be "0". For each and every other transmitted and stored picture, PN shall be increased by 1 (within a given scalability layer, if Annex O is in use). If the difference (modulo 1024) of the PNs of two consecutively received and stored pictures is not 1, the decoder should infer a loss of pictures or corruption of data. In such a case, a back-channel message indicating the loss of pictures may be sent to the encoder.

Besides the PN, each picture stored in the multi-picture buffer has an associated index, called the default relative index. When a picture is first added to the multi-picture buffer it is given default relative index 0 – unless it is assigned to a long-term index. The default relative indices of pictures in the multi-picture buffer are modified when pictures are added to or removed from the multi-picture buffer, or when short-term pictures are assigned to long-term indices.

The pictures stored in the multi-picture buffers can also be divided into two categories: long-term pictures and short-term pictures. A long-term picture can stay in the multi-picture buffer for a long time (more than 1023 coded and stored picture intervals). The current picture is initially considered a short-term picture. Any short-term picture can be changed to a long-term picture by assigning it a long-term index according to information in the bitstream. The PN is the unique ID for all short-term pictures in the multi-picture buffer. When a short-term picture is changed to a long-term picture, it is also assigned a long-term picture index (LPIN). A long-term picture index is assigned to a picture by associating its PN to an LPIN. Once a long-term picture index has been assigned to a picture, the only potential subsequent use of the long-term picture's PN within the bitstream shall be in a repetition of the long-term index assignment. The PNs of the long-term pictures are unique within 1024 transmitted and stored pictures. Therefore, the PN of a long-term picture cannot be used for assignment of a long-term index after 1023 transmitted subsequent stored pictures. LPIN becomes the unique ID for the life of a long-term picture.

PN (for a short-term picture) or LPIN (for a long-term picture) can be used to re-map the pictures into re-mapped relative indices for efficient reference picture addressing.

U.4.1 Decoder Process for Short/Long-term Picture Management

The decoder may have both long-term pictures and short-term pictures in its multi-picture buffer. The MLIP1 field is used to indicate the maximum long-term picture index allowed in the buffer. If no prior value of MLIP1 has been sent, no long-term pictures shall be in use, i.e. MLIP1 shall initially have an implied value of "0" upon invocation of the ERPS mode. Upon receiving an MLIP1 parameter, a new MLIP1 shall take effect until another value of MLIP1 is received. Upon receiving a new MLIP1 parameter in the bitstream, all long-term pictures with associated long-term indices greater than or equal to MLIP1 shall be considered marked "unused". The frequency of transmitting MLIP1 is out of the scope of this Recommendation. However, the encoder should send an MLIP1 parameter upon receiving an error message, such as an Intra request message.

A short-term picture can be changed to a long-term picture by using an MMCO command with an associated DPN and LPIN. The short-term picture number is derived from DPN and the long-term picture index is LPIN. Upon receiving such an MMCO command, the decoder shall change the short-term picture with PN indicated by DPN to a long-term picture and shall assign it to the long-term index indicated by LPIN. If a long-term picture with the same long-term index already exists in the buffer, the previously-existing long-term picture shall be marked "unused". An encoder shall not assign a long-term index greater than MLIP1-1 to any picture. If LPIN is greater than MLIP1-1, this condition should be treated by the decoder as an error. For error resilience, the encoder may send the same long-term index assignment operation or MLIP1 specification message repeatedly. If the picture specified in a long-term assignment operation is already associated with the required LPIN, no action shall be taken by the decoder. An encoder shall not assign the same picture to more than one long term index value. If the picture specified in a long-term index assignment operation is already associated with a different long-term index, this condition should be treated as an error. An encoder shall only change a short-term picture to a long-term picture within 1024 transmitted consecutive stored pictures. In other words, a short-term picture shall not stay in the short-term buffer after more than 1023 subsequent stored pictures have been transmitted. An encoder shall not assign a long-term index to a short-term picture that has been marked as "unused" by the decoding process prior to the first such assignment message in the bitstream. An encoder shall not assign a long-term index to a picture number that has not been sent.

U.4.2 Decoder Process for Reference Picture Buffer Mapping

The decoder employs indices when referencing a picture for motion compensation on the macroblock layer using the fields PR₀, PR, PR₂, PR₃, PR₄, PRB, PRFW, and BSBBW. In pictures other than B pictures, these indices are the default relative indices of pictures in the multi-picture buffer when the fields ADPN and LPIR are not present in the current picture, GOB, or slice layer as applicable, and are re-mapped relative indices when these fields are present. In B pictures, the first one or two pictures (depending on BTPSM) in relative index order are used for backward prediction, and the forward picture reference parameters specify a relative index into the remaining pictures for use in forward prediction.

The indices of pictures in the multi-picture buffer can be re-mapped onto newly specified indices by transmitting the RMPNI, ADPN, and LPIR fields. RMPNI indicates whether ADPN or LPIR is present. If ADPN is present, RMPNI specifies the sign of the difference to be added to a picture number prediction value. The ADPN value corresponds to the absolute difference between the PN of the picture to be re-mapped and a prediction of that PN value. The first transmitted ADPN is computed as the absolute difference between the PN of the current picture and the PN of the picture to be re-mapped. The next transmitted ADPN field represents the difference between the PN of the previous picture that was re-mapped using ADPN and that of another picture to be re-mapped. The process continues until all necessary re-mapping is complete. The presence of re-mappings specified using LPIR does not affect the prediction value for subsequent re-mappings using ADPN. If RMPNI indicates the presence of an LPIR field, the re-mapped picture corresponds to a long-term picture with a long-term index of LPIR. If any pictures are not re-mapped to a specific order by RMPNI, these remaining pictures shall follow after any pictures having a re-mapped order in the indexing scheme, following the default order amongst these non-re-mapped pictures.

If the decoder detects a missing picture, it may invoke some concealment process, and may insert an error-concealed picture into the multi-picture buffer. Missing pictures can be identified if one or several picture numbers are missing or if a picture not stored in the multi-picture buffer is indicated in a transmitted ADPN or LPIR. Concealment may be conducted by copying the closest temporally preceding picture that is available in the multi-picture buffer into the position of the missing picture. The temporal order of the short-term pictures in the multi-picture buffer can be inferred from their default relative index order and PN fields. In addition or instead, the decoder may send a forced INTRA update signal to the encoder by external means (for example, Recommendation H.245), or the decoder may use external means or back-channel messages (for example, Recommendation H.245) to indicate the loss of pictures to the encoder. A concealed picture may be inserted into the multi-picture buffer when using the "Sliding Window" buffering type. If a missing picture is detected when decoding a GOB or Slice layer, the concealment may be applied to the picture as if the missing picture had been detected at the picture layer.

U.4.3 Decoder Process for Sub-Picture Removal

Sub-Picture Removal may be used to reduce the amount of memory required to save multiple reference pictures. In sub-picture removal operation, each reference picture is partitioned into smaller equal-sized sub-pictures. The memory reduction is accomplished by marking undesired sub-picture areas as "unused". The strategy used by the encoder to decide which of the sub-pictures to mark as "unused" is outside the scope of this document. The encoder signals to the decoder the size of the sub-pictures and which of the sub-pictures to mark as "unused" using MMCO commands in the enhanced reference picture selection (ERPS) layer. The encoder shall not send information in the bitstream that causes any samples in reference pictures or sub-pictures that it has caused to be marked as "unused" to be indicated for use in the prediction of subsequent pictures.

The sub-picture removal capability is negotiated by external means (for example, Recommendation H.245). In addition, the decoder signals, also by external means, the minimum partition unit (MPU) which is described in terms of a minimum width and height (in units of 16 luminance samples) of a sub-picture and the total amount of memory it has available for its multi-picture buffer. Memory management is facilitated by the partition rules described below.

Each reference picture is partitioned into rectangular sub-pictures of equal size. The encoder specifies the sub-picture size which shall be an integer multiple of the MPU. The width and height of the sub-picture shall be integer multiples of the minimum MPU width and height negotiated externally. The upper-left-hand corner of the first sub-picture is coincident with the upper-left-hand corner of the reference picture. Consequently, the entire partition may be described by specifying the width and height of a sub-picture. If the picture size is not an integer multiple of the sub-picture size, some sub-pictures may extend beyond the right and bottom boundaries of the reference picture if the picture size is not an integer multiple of the sub-picture size. When a sub-picture that extends past the reference picture boundary is saved, a convenient memory management strategy is to set aside enough memory to save the entire sub-picture, rather than just the memory necessary to save the portion of the reference picture that lies within that sub-picture. This is the convention which shall be followed in any calculation of buffer spare capacity for the purpose of determining buffer fullness (e.g. in order to determine whether to automatically mark buffered pictures as "unused" in "sliding window" operation). A decoder designed such that each sub-picture occupies the same amount of memory will prevent the possibility of memory fragmentation.

An example method designed to access referenced picture samples when sub-picture removal is in use is described briefly as follows. One important element in any reference picture access technique is a mechanism to identify where the samples in each sub-picture are stored in memory. If there are R reference pictures and each picture is partitioned into S sub-pictures then there are a total of K = R·S sub-pictures. For example, the sub-picture in the upper-left hand corner of the first reference picture number can be considered sub-picture number 0, and the sub-picture to the right of it can be considered sub-picture number 1

and so on in raster scan order progressing from reference picture 0 to $R-1$ until all K sub-pictures have a label. The total buffer capacity is $SPTN$ sub-picture memory buffers, and $SPTN$ is typically less than K . A K -element array can be defined, $subPicMem[K]$, such that $t = subPicMem[k]$ corresponds to the sub-picture memory area that contains the samples in sub-picture k . For example, a case can be considered in which $R = 5$ reference pictures each have $S = 12$ sub-pictures. Then the samples for the sub-picture 6 in reference picture 3 would be found in sub-picture memory area $t = subPicMem[k]$ where $k = 3 \cdot S + 6 = 42$.

For example, when referencing samples for motion-compensated prediction of one block of luminance or chrominance data when the Advanced Prediction and Reduced Resolution Update optional modes are not in use, it is necessary to acquire $n \times m$ samples, where n and m may take values of 8 or 9 to accommodate half-integer motion compensation. Since the samples in one block may lie in up to four different sub-pictures, four separate cases must be considered. In all cases, the first step is to find the location in memory that contains the upper-left hand sample (U) of the block to be referenced. The sub-picture containing U can be identified by dividing the horizontal or vertical location of U by the sub-picture width or height. If U lies in sub-picture k , then that sample will be located in the $subPicMem[k]$ sub-picture memory area. Next, if both the sample $m-1$ samples to the right of U (i.e. the upper-right-hand corner of the block) and the sample $n-1$ samples down from U (i.e. the lower-left-hand corner of the block) lie in sub-picture k , this can be considered case number one. If the sample $n-1$ samples down from U lies within k , but the sample $m-1$ samples to the right of U does not, this can be considered case two. If the sample $m-1$ samples to the right of U lies within k , but the sample $n-1$ down does not, this can be considered case three. Otherwise, when both the sample $m-1$ samples to the right of U and the one $n-1$ samples down lie outside of sub-picture k , this can be considered case four.

In case number one, all samples in the reference block are contained within the k^{th} sub-picture. In this case, all relevant $n \times m$ samples may be found in sub-picture memory area $subPicMem[k]$. In case two, the samples that lie in the k^{th} sub-picture can be obtained from sub-picture memory area $subPicMem[k]$ and the remaining samples can be obtained from $subPicMem[k_r]$ where k_r is the sub-picture to the right of k . In case three, the samples that lie in the k^{th} sub-picture can be obtained from memory area $subPicMem[k]$, and the remaining samples can be obtained from $subPicMem[k_d]$ where k_d is the sub-picture below k . In case four, the samples that lie in the k^{th} sub-picture can be obtained from sub-picture memory area $subPicMem[k]$ and the remaining samples can be obtained from memory areas $subPicMem[k_r]$, $subPicMem[k_d]$ and $subPicMem[k_{rd}]$ where k_r and k_d are defined above and k_{rd} is the sub-picture to the right and below k .

U.4.4 Decoder Process for Multi-Picture Motion Compensation

Multi-picture motion compensation is applied if the MRPA field indicates the use of more than one reference picture. For multi-picture motion compensation, the decoder chooses a reference picture as indicated using the fields PR_0 , PR , PR_2 , PR_3 , PR_4 , PRB , $PRFW$, $PRBW$, and $BSBBW$ on the macroblock layer. Once, the reference picture is specified, the decoding process for motion compensation proceeds as described in subclause 6.1.

In case four motion vectors per macroblock are used and the MRPA field indicates the use of more than one reference picture, the picture reference index for both chrominance blocks is that associated with the first of the four motion vectors (with the motion compensation process otherwise as specified by subclause 6.1).

U.4.5 Decoder Process for Reference Picture Buffering

The buffering of the currently decoded picture can be specified using the reference picture buffering type (RPBT) for non-B pictures. The buffering may follow a first-in, first-out ("Sliding Window") mode. Alternatively, the buffering may follow a customized adaptive buffering ("Adaptive Memory Control") operation that is specified by the encoder in the forward channel. B pictures do not affect buffer contents.

The "Sliding Window" buffering type operates as follows. First, the decoder determine whether the picture can be stored into "unused" buffer capacity. If there is insufficient "unused" buffer capacity, the short term picture with the largest default relative index (i.e. the oldest short-term picture in the buffer) shall be marked as "unused". This process is repeated if necessary (in the case of sub-picture removal) until sufficient memory capacity is freed to hold the current decoded picture. The current picture is stored in the buffer and assigned a default relative index of zero. The default relative index of all other short-term pictures is incremented by one. The default relative indices of all long-term picture are incremented by one minus the number of short-term pictures removed.

In the "Adaptive Memory Control" buffering type, specified pictures or sub-picture areas may be removed from the multi-picture buffer explicitly. The currently decoded picture, which is initially considered a short-term picture, may be inserted into the buffer with default relative index 0, may be assigned to a long-term index, or may be marked as "unused" by the encoder. Other short-term pictures may also be assigned to long-term indices. The buffering process shall operate in a manner functionally equivalent to the following: First, the current picture is added to the multi-picture buffer with default relative index 0, and the default relative indices of all other pictures are incremented by one. Then, the MMCO commands are processed:

- If MMCO indicates a reset of the buffer contents by using RESET equal to "1", all pictures in the buffer are marked as "unused" except the current picture (which will be the picture with default relative index 0 since a buffer reset must be the first MMCO command as required by subclause U.3.1.5.7).
- If MMCO indicates a maximum long-term index using MLIP1, all long-term pictures having long-term indices greater than or equal to MLIP1 are marked as "unused" and the default relative index order of the remaining pictures are not affected.
- If MMCO indicates that a picture is to be marked as "unused" in the multi-picture buffer and if that picture has not already been marked as "unused", the specified picture is marked as "unused" in the multi-picture buffer and the default relative indices of all subsequent pictures in default order are decremented by one.
- If MMCO indicates that sub-picture areas of some picture are to be marked as "unused" in the multi-picture buffer, the specified sub-picture areas are marked as "unused" and the default relative index order of the pictures is not affected. As required by subclause, U.3.1.5.10, not all sub-picture areas of any given picture will be marked "unused" by a sub-picture removal MMCO command (instead, the encoder should send an MMCO command marking the picture as a whole as "unused").
- If MMCO indicates the assignment of a long-term index to a specified short-term picture and if the specified long-term index has not already been assigned to the specified short-term picture, the specified short-term picture is marked in the buffer as a long-term picture with the specified long-term index. If another picture is already present in the buffer with the same long-term index as the specified long-term index, the other picture is marked as "unused". All short-term pictures that were subsequent to the specified short-term picture in default relative index order and all long-term pictures having a long-term index less than the specified long-term index have their associated default relative indices decremented by one. The specified picture is assigned to a default relative index of one plus the highest of the incremented default relative indices, or zero if there are no such incremented indices.

The resulting buffered quantity of pictures or sub-picture regions not marked as "unused" shall not exceed the buffer capacity indicated by the most recent value of SPTN. If the decoder detects this condition, it should be treated as an error.

U.5 Back-Channel Messages

An out-of-band channel, which need not necessarily be reliable, can be used to convey back-channel messages. The syntax of this out-of-band channel (which could be a separate logical channel, for example using Recommendation H.223 or Recommendation H.225.0) should be the one defined herein. The "videomux" operation of back-channel messages as defined in Annex N is not supported in the ERPS mode.

U.5.1 BCM Separate Logical Channel Layer

The BCM layer as specified in subclause U.5.2 should be carried by a BCM Separate Logical Channel layer as shown in Figure U.9.

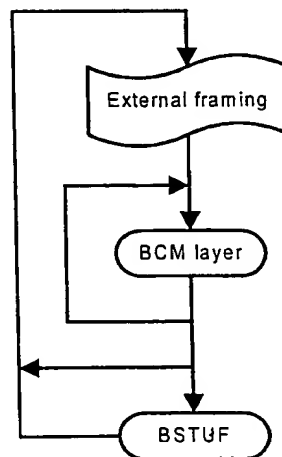


FIGURE U.9/H.263

Structure of BCM Separate Logical Channel layer for ERPS mode

U.5.1.1 External Framing

External framing of back-channel messages should be provided as shown in Figure U.9. The external framing is used to determine the starting point for the back-channel messages and the amount of back-channel message data to follow.

U.5.1.2 Back-Channel Stuffing (BSTUF) (variable length)

BSTUF is a variable length codeword that may be present only after the last back-channel message in an external frame. BSTUF consists of a codeword of variable length consisting of one or more bits of value "0".

U.5.2 Back-Channel Message Layer Syntax

The syntax for the back-channel message (BCM) layer defined herein shall be as shown in Figure U.10.

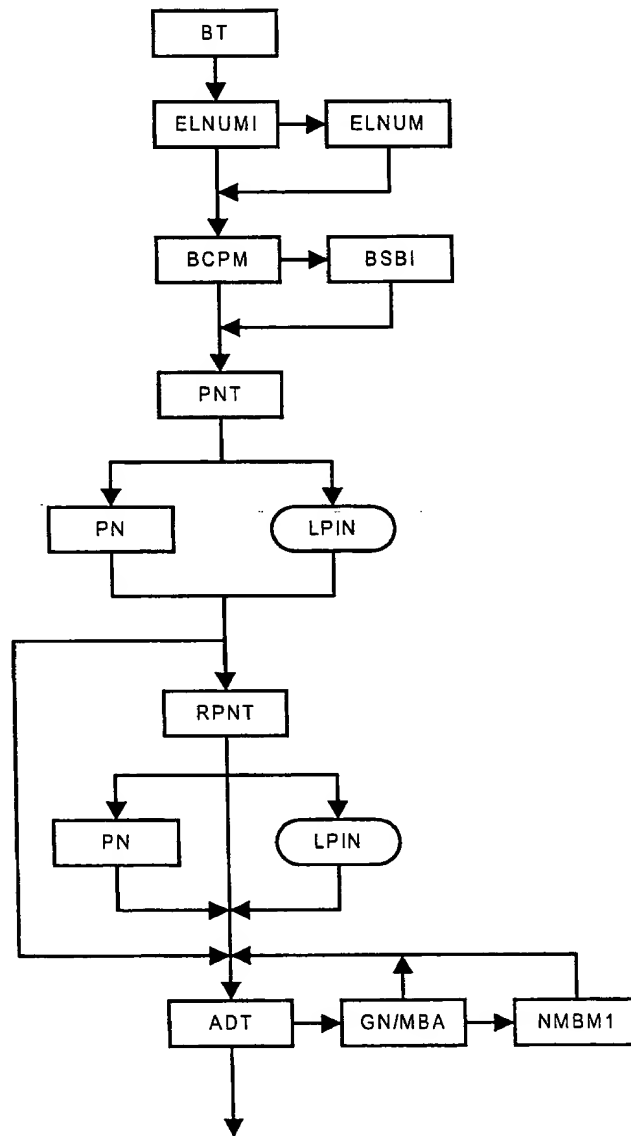


FIGURE U.10/H.263

Structure of Back-Channel Message (BCM) layer for ERPS mode

U.5.2.1 Back-Channel Message Type (BT) (2 bits)

BT is a two bit fixed length codeword which indicates the type of back-channel message. BT is the first codeword present in each back-channel message. Which type or types of message are requested by the encoder is indicated in the RPSMF field of the forward-channel syntax. The values of BT shall be defined as:

- "00": Reserved for future use,
- "01": Reserved for future use,
- "10": NACK: This indicates the loss or erroneous decoding of the corresponding part of the forward channel data,
- "11": ACK: This indicates the correct decoding of the corresponding part of the forward channel data.

U.5.2.2 Enhancement Layer Number Indication (ELNUMI) (1 bit)

ELNUMI is a single bit fixed length codeword that follows BT in the back-channel message. ELNUMI shall be "0" unless the optional Temporal, SNR, and Spatial Scalability mode (see Annex O) is used in the forward channel and some enhancement layers of the forward channel are combined in one logical channel and the back-channel message refers to an enhancement layer (rather than the base layer), in which case ELNUMI shall be "1".

U.5.2.3 Enhancement Layer Number (ELNUM) (4 bits)

ELNUM is a 4 bit fixed length codeword that is present only if ELNUMI is "1". It follows ELNUMI if present. When present, ELNUM contains the layer number of the enhancement layer referred to in the back-channel message.

U.5.2.4 Back-Channel CPM Indicator (BCPM) (1 bit)

BCPM is a single bit fixed length codeword that follows ELNUMI or ELNUM in the back-channel message. BCPM shall be "0" unless the CPM mode (see subclause 5.2.4 and Annex C) is used in the forward channel data, in which case BCPM shall be "1". If BCPM is "1", this indicates that BSBI is present.

U.5.2.5 Back-Channel Sub-Bitstream Indicator (BSBI) (2 bits)

BSBI is a 2 bit fixed length codeword that follows BCPM when present. BCPM is present only if BCPM is "1". BSBI is the natural binary representation of the Sub-Bitstream number in the forward channel data to which the back-channel message refers (see subclause 5.2.4 and Annex C).

U.5.2.6 Picture Number Type (PNT) (1 bit)

PNT is a single bit fixed length codeword that is always present and follows BCPM or BSBI in the back-channel message. The values of PNT shall be defined as:

- "0": The message concerns a picture specified by a short-term picture number (PN),
- "1": The message concerns a picture specified by a long-term picture index (LPIN).

PNT is followed by PN or LPIN, depending on the value of PNT. PN and LPIN shall be represented as specified for use in forward channel data in subclauses U.4.1.3 and U.4.1.5.9, respectively.

U.5.2.7 Requested Picture Number Type (RPNT) (2 bits)

RPNT is a 2 bit fixed length codeword that is present only if BT indicates a NACK message. It follows PN or LPIN when present. It determines how to identify a picture in the multi-picture buffer which may be used as a reference for the coding of subsequent pictures. The values of RPNT shall be defined as:

- "00": No valid pictures in buffer – buffer should be reset by an I or EI picture with RESET equal to "1",
- "01": No particular picture is identified to be used as a reference,
- "10": A picture which may be used as a reference is identified by a short-term picture number (PN),
- "11": A picture which may be used as a reference is identified by a long-term picture index (LPIN).

If RPNT is "10" or "11", RPNT is followed by PN or LPIN, depending on the value of RPNT. PN and LPIN shall be represented as specified for use in forward channel data in subclauses U.4.1.3 and U.4.1.5.9, respectively. Typically the PN or LPIN specified using RPNT identifies the last correctly decoded spatially-corresponding picture area for the picture or region identified in the back-channel message.

U.5.2.8 Additional Data Type (ADT) (2 bits)

ADT is a 2 bit fixed length codeword that is present after PN, LPIN, or RPNT, as determined by PNT (in an ACK message) or RPNT (in a NACK message). It may occur multiple times if present. It specifies the type of additional data used to identify a region of the picture of concern to which the back-channel message applies. The values of ADT shall be defined as:

"00": End of additional data,

"01": A region is identified by only a GN/MBA field,

"10": A region is identified as a raster-scan area within a picture by GN/MBA and NMBM1,

"11": A region is identified as a raster-scan area within a rectangular slice by GN/MBA and NMBM1.

If ADT is "00", no more data follows in the back-channel message. If ADT is "01", ADT is followed by GN/MBA and then by another ADT. If ADT is "10" or "11", ADT is followed by GN/MBA and NMBM1 and then by another ADT.

If ADT is "10", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within the picture. If ADT is "11", the region is identified as a region starting at a particular spatial location specified by GN/MBA and containing a specified number of macroblocks in raster-scan order within a rectangular slice. If ADT is present only once and is "00", the region identified is the picture as a whole. If ADT is present more than once, the value "00" is used only to end the loop rather than to identify a region.

U.5.2.9 GOB Number/Macroblock Address (GN/MBA) (5/6/7/9/11/12/13/14 bits)

GN/MBA is a fixed length codeword which specifies a GOB number or macroblock address. GN/MBA follows ADT when present. GN/MBA is present when indicated by ADT. If the optional Slice Structured mode (see Annex K) is not in use, GN/MBA contains the GOB number of the beginning of an area to which the back-channel message refers. If the optional Slice Structured mode is in use, GN/MBA contains the macroblock address of the beginning of the area to which the back-channel message refers. The length of this field shall be as specified elsewhere in this Recommendation for GN or MBA.

U.4.5.2.10 Number of Macroblocks Minus 1 (NMBM1) (5/6/7/9/11/12/13/14 bits)

NMBM1 is a fixed length codeword which specifies a number of macroblocks. NMBM1 is present when indicated by ADT. It follows GN/MBA when present. It contains the natural representation of the number of specified macroblocks minus 1. The length of this field shall be the length defined for a macroblock address in subclause K.2.5 and Table K.2.

Annex V Data Partitioned Slice Mode

(This annex forms an integral part of this Recommendation.)

V.1 Introduction

This annex describes the optional data-partitioned slice (DPS) mode of H.263. The capability of this mode is signaled by external means (for example Recommendation H.245). The use of this mode shall be indicated by setting the formerly-reserved bit 17 of the optional part of the PLUSPTYPE (OPPTYPE) to '1'. This mode uses the header structure defined in Annex K.

Data partitioning provides robustness in error prone environments. This is accomplished using a rearrangement of the H.263 syntax to enable early detection of and recovery from errors that have been introduced during transmission.

V.2 Structure of data partitioning

When data partitioning is used, the data is arranged as a video picture segment, as defined in Section R.2. The MB's in the segment are rearranged so that the header information for all the MB's in the segment are transmitted together, followed by the MV's for all the MB's in the segment, and then by the DCT coefficients for all the MB's in the segment. The segment header uses the same syntax as described in Section K.2. The header, MV, and DCT partitions are separated by markers, allowing for resynchronization at the end of the partition in which an error occurred. Each segment shall contain the data for an integer number of MB's. When this mode is in use the syntax shown in Figure V.1 shall be used.

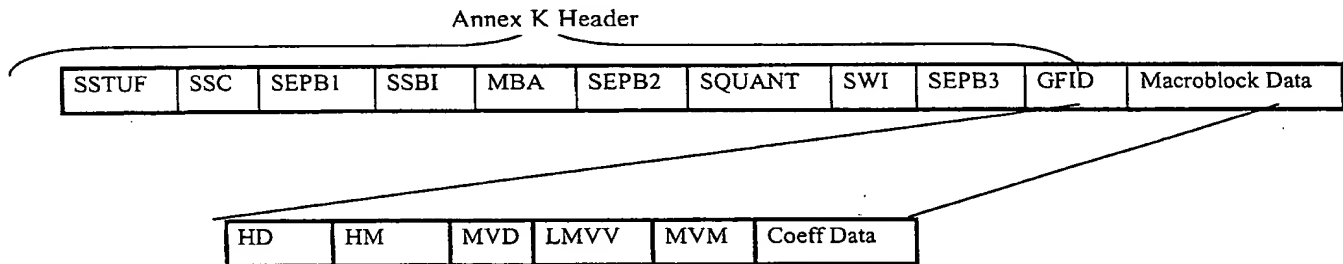


FIGURE V.1/H.263

Data Partitioning Syntax

Note that when this annex is not active, the MV and DCT data are transmitted in an interleaved fashion for all the MB's in a video picture segment, in which case an error normally results in the loss of all information for the remaining MB's in the packet.

V.2.1 Header Data (HD) (Variable length)

The Header Data field contains the COD and MCBPC information for all the MB's in the packets, plus the MODB data in case of PB-frames or Improved PB-frames. A reversible variable length code (RVLC) is used to combine the COD and the MCBPC for all the MB's in the packet. This code is shown in Tables V.1 through V.5/H.263. If Annex O is in use, the COD is only combined with the MB TYPE to form the RVLC for B and EP pictures using tables V.3 and V.4, and the CBPC is coded with codewords in Table O.4. If COD=0 and Annex G or Annex M is in use, the codeword for the COD+MCBPC shall be immediately followed by the reversible variable-length encoded data corresponding to the MODB field of the macroblock. Table V.6 shall be used for PB-frames, Table V.7 shall be used for Improved PB-frames.

V.2.2 Header Marker (HM) (9 bits)

A codeword of 9 bits. Its value is 1010 0010 1. The HM terminates the header partition. When reversed decoding is used by a decoder, the decoder searches for this marker. This value cannot occur naturally in the HD field.

V.2.3 Motion Vector Data Layer (Variable length)

V.2.3.1 Motion Vector Difference Coding

For the motion vectors, the RVLC codewords shown in Table D.3/H.263 are used to encode the difference between the motion vector and the motion vector prediction. Note that this annex only uses the entropy coding from Annex D, but not the other aspects of it unless Annex D is also in use.

V.2.3.2 Prediction of Motion Vector Values

The first motion vector in the packet is coded using a predictor value of 0 for both horizontal and vertical components, and the MV's for the subsequent coded MB's are coded predictively using the MV difference (MVD). This differs from the method otherwise used for coding the MV's in which the MV's following a skipped or INTRA MB are coded using a predictor value of 0 for both horizontal and vertical components.

Forward Direction: $MV_i = MV_{i-1} + MVD_i = MV_{i-1} + (MV_i - MV_{i-1})$

Backward Direction: $MV_{i-1} = MV_i - MVD_i = MV_i - (MV_i - MV_{i-1})$.

(MV_i and MVD_i are the i th MV and MV Difference in the packet respectively)

The motion vector information for the last motion vector in the packet is coded in this manner and is also coded again in the LMVV field as described below in V.2.4. This allows the decoder to independently decode the sequence of MV's using two different prediction paths: 1) in the forward direction, starting from the beginning of the motion data of the packet, and 2) in the backward direction, from the end of the motion data in a packet. This provides robustness for better error detection and concealment.

NOTE: When the DPS mode is not in use, motion vectors are predictively coded, with the prediction of the current motion vector being the median value of 3 motion vectors of neighboring locations as described in Section 6.1.1. Because packets in this annex are formed in a way such that the number of MB's coded in each packet is variable, using the median predictive coding method (which involves motion vectors on different rows of the frame) would prevent reversible decoding of the motion vectors in a slice. When the DPS mode is in use, a single prediction thread is formed for the MV's in the whole packet. This is shown in Figure V.2.

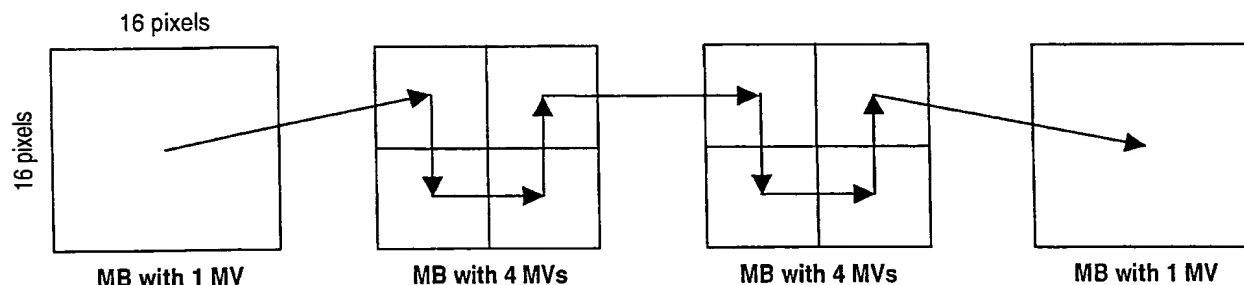


FIGURE V.2/H.263

Single Thread Motion Vector Prediction

In case of B pictures or EP pictures (Annex O), MVDFW and MVDBW may be present as indicated by the MBTYPE codeword in tables V.3 and V.4. MVDFW is predictively encoded using the same single prediction thread as described above and MVDBW (when present in B pictures) shall be encoded as specified in O.4.6. MVDFW and MVDBW shall be coded with the codewords from Table D.3/H.263.

In case of PB-frames (Annex G) and Improved PB-frames (Annex M), the MVDB data shall be encoded as specified in corresponding annexes and shall be coded using the codewords from Table D.3/H.263.

NOTE – If the backward decoding mode is engaged in a B frame (Annex O) or in Improved PB-frames (Annex M), MVDB and MVDBW should be discarded by the decoder as the Motion Vector data for the backward prediction may not be recovered properly across the packet boundaries.

V.2.3.3 Start-Code Emulation Prevention in Motion Vector Difference Coding

The MVD start-code-emulation avoidance method is changed from the method described in Section D.2 of Annex D, in order to facilitate independent parsing in the backward direction. A $MVD=0$ (codeword "1") shall be inserted between any two consecutive MVD's that are both equal to 1 (codeword "000"). This differs from Annex D, in which the bit is only inserted when two consecutive $MVD=1$ form a pair (i.e. when the first MVD is the horizontal component, and the second is the vertical component). If Annex D and Annex V are both in use, this Annex V method of start-code-emulation avoidance method shall be used instead of the method described in Section D.2.

V.2.4 Last Motion Vector Value (LMVV) (Variable length)

The LMVV field contains the last MV in the packet. It is coded using a predictor value of 0 for both the horizontal and vertical components. If there are no motion vectors or only one motion vector in the packet, LMVV shall not be present. (This use of a fixed zero-valued predictor enables the use of reversible decoding.)

V.2.5 Motion Vector Marker (MVM) (10 bits)

A codeword of 10 bits having the value '0000 0000 01'. The MVM terminates the motion vector partition. When reverse decoding is used in a decoder, the decoder searches for this marker. The Motion Vector Marker (MVM) shall not be included in the packet if the packet does not contain Motion Vector Data (if all the macroblocks in the packet are intra-coded or with COD's equal to 1).

V.2.6 Coefficient Data Layer (Variable length)

The DCT data layer contains INTRA_MODE (if present), CBPB (if present), CBPC (if present), CBPY, DQUANT (if present), and DCT coefficients coded as specified in Sections I.2, 5.3.4, O.4.3, 5.3.5, 5.3.6, and 5.4.2, respectively. The syntax diagram of DCT Data is illustrated in Figure V.3. The presence of CBPC is indicated in tables V.3 and V.4.

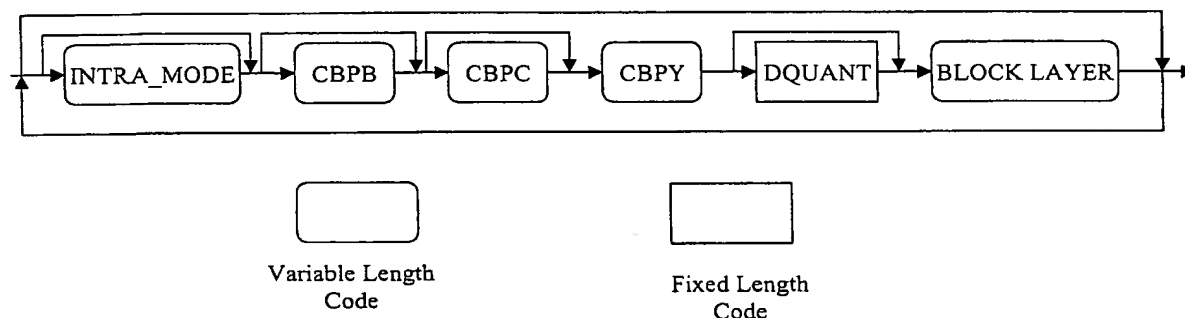


FIGURE V.3/H.263

Coefficient Data syntax

V.3 Interaction with Other Optional Modes

The DPS mode acts effectively as a sub-mode of the Slice Structured mode of Annex K, and uses its outer picture and slice header structures. The SS mode shall therefore be indicated as being in use whenever the DPS mode is in use. Both of the other sub-modes of the Slice Structured mode (the Arbitrary Slice Ordering and Rectangular Slice sub-modes) may be used in conjunction with the DPS mode.

The Syntax-Based Arithmetic Coding mode of Annex E shall not be used with this annex, as it does not allow for reversible decoding.

Annex H Forward Error Correction should not be used with this annex, as it can result in the bitstream being disrupted in undesirable places. However, the use of Annex H with the DPS mode is not forbidden, as the FEC defined in Annex H is required in some existing standard system designs.

The Temporal, SNR, and Spatial Scalability (TSSS) mode of Annex O may be used in conjunction with the DPS mode. When the TSSS and DPS modes are used together, the codewords provided in Tables V.3, V.4, and V.5 shall be used instead of those defined in Annex O.

Annex U shall not be used with this Annex.

TABLE V.1/H.263

COD + MCBPC RVLC table for INTRA MB's

TABLE V.1/H.263

COD + MCBPC RVLC table for INTRA MB's

| MB type | CBPC (56) | Codeword (for combined COD+MCBPC) | Number of Bits |
|-------------|-----------|---|----------------|
| 3 (INTRA) | 00 | 1 | 1 |
| 3 | 01 | 010 | 3 |
| 3 | 10 | 0110 | 4 |
| 3 | 11 | 01110 | 5 |
| 4 (INTRA+Q) | 00 | 00100 | 5 |
| 4 | 01 | 011110 | 6 |
| 4 | 10 | 001100 | 6 |
| 4 | 11 | 0111110 | 7 |
| stuffing | | 0011100 | 7 |

TABLE V.2/H.263

COD + MCBPC RVLC Table for INTER MB's

| MB type | CBPC (56) | Codeword (for combined COD+MCBPC) | Number of Bits |
|-----------------|-----------|---|----------------|
| skipped | | 1 | 1 |
| 0 (INTER) | 00 | 010 | 3 |
| 0 | 10 | 00100 | 5 |
| 0 | 01 | 011110 | 6 |
| 0 | 11 | 0011100 | 7 |
| 1 (INTER + Q) | 00 | 01110 | 5 |
| 1 | 10 | 00011000 | 8 |
| 1 | 01 | 011111110 | 9 |
| 1 | 11 | 0111111110 | 11 |
| 2 (INTER4V) | 00 | 0110 | 4 |
| 2 | 10 | 01111110 | 8 |
| 2 | 01 | 00111100 | 8 |
| 2 | 11 | 000010000 | 9 |
| 3 (INTRA) | 00 | 001100 | 6 |
| 3 | 11 | 0001000 | 7 |
| 3 | 10 | 001111100 | 9 |
| 3 | 01 | 000111000 | 9 |
| 4 (INTRA + Q) | 00 | 0111110 | 7 |
| 4 | 11 | 0011111100 | 10 |
| 4 | 10 | 0001111000 | 10 |
| 4 | 01 | 0000110000 | 10 |
| 5 (INTER4V + Q) | 00 | 00111111100 | 11 |
| 5 | 01 | 00011111000 | 11 |
| 5 | 10 | 00001110000 | 11 |
| 5 | 11 | 00000100000 | 11 |
| stuffing | | 0111111110 | 10 |

TABLE V.3/H.263

MBTYPE RVLC codes for B MB's

| Index | Prediction Type | MVDFW | MVDBW | CBPC + CBPY | DQUANT | MBTYPE | Bits |
|-------|-----------------------|-------|-------|----------------|--------|-----------|------|
| — | Direct (skipped) | | | | | 1 (COD=1) | 1 |
| 0 | Direct | | | X | | 010 | 3 |
| 1 | Direct + Q | | | X | X | 001100 | 6 |
| 2 | Forward (no texture) | X | | | | 00100 | 5 |
| 3 | Forward | X | | X | | 011110 | 6 |
| 4 | Forward + Q | X | | X | X | 01111110 | 8 |
| 5 | Backward (no texture) | | X | | | 0110 | 4 |
| 6 | Backward | | X | X | | 01110 | 5 |
| 7 | Backward + Q | | X | X | X | 00111100 | 8 |
| 8 | Bi-Dir (no texture) | X | X | | | 0011100 | 7 |
| 9 | Bi-Dir | X | X | X | | 0001000 | 7 |
| 10 | Bi-Dir + Q | X | X | X | X | 0111110 | 7 |
| 11 | INTRA | | | X | | 00011000 | 8 |
| 12 | INTRA + Q | | | X | X | 011111110 | 9 |
| 13 | Stuffing | | | | | 001111100 | 9 |

TABLE V.4/H.263

MBTYPE RVLC Table for EP MB's

| Index | Prediction Type | MVDFW | MVDBW | CBPC + CBPY | DQUANT | MBTYPE | Bits |
|-------|---------------------|-------|-------|----------------|--------|-----------|------|
| — | Forward (skipped) | | | | | 1 (COD=1) | 1 |
| 0 | Forward | X | | X | | 010 | 3 |
| 1 | Forward + Q | X | | X | X | 0110 | 4 |
| 2 | Upward (no texture) | | | | | 01110 | 5 |
| 3 | Upward | | | X | | 00100 | 5 |
| 4 | Upward + Q | | | X | X | 011110 | 6 |
| 5 | Bi-Dir (no texture) | | | | | 001100 | 6 |
| 6 | Bi-Dir | X | | X | | 0111110 | 7 |
| 7 | Bi-Dir + Q | X | | X | X | 0011100 | 7 |
| 8 | INTRA | | | X | | 0001000 | 7 |
| 9 | INTRA + Q | | | X | X | 01111110 | 8 |
| 10 | Stuffing | | | | | 00111100 | 8 |

TABLE V.5/H.263

COD + MCBPC RVLC Table for EI MB's

| Prediction type | QCBP (56) | Codeword (for combined COD+MCBPC) | Number of Bits |
|------------------|-----------|---|----------------|
| Upward (skipped) | | 1 | 1 |
| 0 (Upward) | 00 | 010 | 3 |
| 0 | 01 | 0110 | 4 |
| 0 | 10 | 01110 | 5 |
| 0 | 11 | 00100 | 5 |
| 1 (Upward + Q) | 00 | 011110 | 6 |
| 1 | 01 | 001100 | 6 |
| 1 | 10 | 0111110 | 7 |
| 1 | 11 | 0011100 | 7 |
| 2 (INTRA) | 00 | 0001000 | 7 |
| 2 | 01 | 01111110 | 8 |
| 2 | 10 | 00111100 | 8 |
| 2 | 11 | 00011000 | 8 |
| 3 (INTRA + Q) | 00 | 011111110 | 9 |
| 3 | 01 | 001111100 | 9 |
| 3 | 10 | 000111000 | 9 |
| 3 | 11 | 000010000 | 9 |
| Stuffing | | 0111111110 | 10 |

TABLE V.6/H.263

RVLC Table for MODB

| Index | CBPB | MVDB | Number of bits | Code |
|-------|------|------|----------------|-------|
| 0 | | | 3 | 010 |
| 1 | | x | 4 | 0110 |
| 2 | x | x | 5 | 01110 |

Note: "x" means that the item is present in the macroblock

TABLE V.7/H.263

RVLC Table for MODB for Improved PB-frames mode

| Index | CBPB | MVDB | Number of bits | Code | Coding Mode |
|-------|------|------|----------------|--------|---------------------------|
| 0 | | | 3 | 010 | Bi-directional prediction |
| 1 | x | | 4 | 0110 | Bi-directional prediction |
| 2 | | x | 5 | 01110 | Forward prediction |
| 3 | x | x | 5 | 00100 | Forward prediction |
| 4 | | | 6 | 011110 | Backward prediction |
| 5 | x | | 6 | 001100 | Backward prediction |

Note — The symbol "x" in the table above indicates that the associated syntax element is present.

Annex W

Additional Supplemental Enhancement Information Specification

(This annex forms an integral part of this Recommendation.)

W.1 Introduction

This annex describes the format of the additional supplemental enhancement information sent in the PSUPP field of the picture layer of H.263, which adds to the functionality defined in Annex L. The capability of a decoder to provide any or all of the capabilities described in this annex may be signaled by external means (for example, Recommendation H.245). Decoders which do not provide the additional capabilities may simply discard any of the newly defined PSUPP information bits that appear in the bitstream. The presence of this supplemental enhancement information is indicated by the presence of both the PEI bit, and by the following PSUPP octet whose FTYPE field has one of the two newly defined values. The basic interpretation of PEI, PSUPP, FTYPE, and DSIZE is identical to Annex L and to sections 5.1.24 and 5.1.25.

W.2 References

The following Recommendations and other references contain provisions which, through reference in this text, constitute provisions of this Recommendation. At the time of publication, the editions indicated were valid. All Recommendations and other references are subject to revision; all users of this Recommendation are therefore encouraged to investigate the possibility of applying the most recent edition of the Recommendations and other references listed below.

- [8] ISO/IEC 10646-1 (1993): Universal Multiple Octet Coded Character Set
- [9] IETF RFC 2396 (1998): Uniform Resource Identifiers (URI): Generic Syntax

W.3 Additional FTYPE Values

Two values that were reserved in Annex L, Table L.1 are defined as follows.

TABLE W.1/H.263

FTYPE Function Type Values

| | |
|----|------------------|
| 13 | Fixed-Point IDCT |
| 14 | Picture Message |

W.4 Recommended Maximum Number of PSUPP Octets

When using any of the aforementioned FTYPE functions defined in this annex, the total number of PSUPP octets per picture should, in relation to the coded picture size, be kept reasonably small, and should not exceed 256 octets regardless of the coded picture size.

NOTE: Some data transmission protocols used for conveyance of the video bitstream may provide for external repetition of picture header contents for error resilience purposes, and may place limits on the amount of such data that can be repeated from a picture header (e.g., 504 bits in the IETF RFC 2429 packetization format). The inclusion of a large number of PSUPP octets may result in the lack of such an external protocol to provide for full repetition of the picture header contents.

W.5 Fixed-Point IDCT

The fixed-point IDCT function indicates that a particular IDCT approximation is used in construction of the bitstream. DSIZE shall be equal to 1 for the fixed-point IDCT function. The octet of PSUPP data that follows specifies the particular IDCT implementation. A value of 0 indicates the reference IDCT 0 as described in W.5.3; values of 1 through 255 are reserved.

W.5.1 Decoder Operation

The capability of a decoder to perform a particular fixed-point IDCT may be signaled to the encoder by external means (for example, Recommendation H.245). When receiving an encoded bitstream with the fixed-point IDCT indication, a decoder shall use the particular fixed-point IDCT if it is capable of doing so.

W.5.2 Removal of Forced Updating

Annex A specifies the accuracy requirements for the inverse discrete cosine transform (IDCT), allowing numerous compliant implementations. To control accumulation of errors due to mismatched IDCTs at the encoder and decoder, Section 4.4 Forced Updating requires that macroblocks be coded in INTRA mode at least once every 132 times when coefficients are transmitted.

If the fixed-point IDCT function type is indicated in the bitstream, then the forced updating requirement is removed, and the frequency of INTRA coding is unregulated. An encoder should continue to use forced updating, however, unless it has ascertained through external means that the decoder is capable of the particular fixed-point IDCT specified herein; otherwise there may be mismatch.

W.5.3 Reference IDCT 0

The reference IDCT 0 is any implementation that, for every input block, produces identical output values as the C source program listed below.

NOTE: This fixed-point IDCT is compliant with Annex A of ITU-T Recommendation H.263, but is not compliant with the extended range of values requirement in Annex A of ITU-T Recommendation H.262 | ISO/IEC 13818-2.

```

/*****
 *
 *                      FIXED-POINT IDCT
 *
 * Fixed-point fast, separable idct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 12 bits signed, stored in 16 bits
 * Output range: [-256, +255]
 * All operations are signed
 *
 *****/

/*
 * Includes
 */

#include <stdlib.h>
#include <stdio.h>

/*
 * Typedefs
 */

typedef short int REGISTER; /* 16 bits signed */
typedef long int LONG; /* 32 bits signed */

/*
 * Global constants
 */

const REGISTER cpo8 = 0x539f; /* 32768*cos(pi/8)*1/sqrt(2) */
const REGISTER spo8 = 0x4546; /* 32768*sin(pi/8)*sqrt(2) */
const REGISTER cpo16 = 0x7d8a; /* 32768*cos(pi/16) */
const REGISTER spo16 = 0x18f9; /* 32768*sin(pi/16) */
const REGISTER c3po16 = 0x6a6e; /* 32768*cos(3*pi/16) */
const REGISTER s3po16 = 0x471d; /* 32768*sin(3*pi/16) */
const REGISTER Oor2 = 0x5a82; /* 32768*1/sqrt(2) */

/*
 * Function declarations
 */

void Transpose(REGISTER block[64]);
void HalfSwap(REGISTER block[64]);
void Swap(REGISTER block[64]);
void Scale(REGISTER block[64], signed char sh);
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max);
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh);
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv);
void Butterfly(REGISTER column[8], char pass);
void IDCT(REGISTER block[64]);

```



```

/*
 * Transpose():
 *   Transpose a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void Transpose(REGISTER block[64])
{
    int i, j;
    REGISTER temp;

    for (i=0; i<8; i++) {
        for (j=0; j<i; j++) {
            temp = block[8*i+j];
            block[8*i+j] = block[8*j+i];
            block[8*j+i] = temp;
        }
    }
    return;
}

/*
 * HalfSwap():
 *   One-dimensional swap
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void HalfSwap(REGISTER block[64])
{
    int i;
    REGISTER temp;

    for (i=0; i<8; i++) {
        temp = block[8+i];
        block[8+i] = block[32+i];
        block[32+i] = temp;
        temp = block[24+i];
        block[24+i] = block[48+i];
        block[48+i] = temp;
        temp = block[40+i];
        block[40+i] = block[56+i];
        block[56+i] = temp;
    }
    return;
}

/*
 * Swap():
 *   Swap and transpose a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void Swap(REGISTER block[64])
{
    HalfSwap(block);
    Transpose(block);
    HalfSwap(block);
}

```

```

/*
 * Scale():
 *   Scale a block
 * Input:
 *   REGISTER block[64]
 *   signed char sh
 * Output:
 *   block
 * Return value:
 *   none
 */
void Scale(REGISTER block[64], signed char sh)
{
    int i;

    if (sh>0) {
        for (i=0; i<64; i++)
            block[i] >>= sh;
    }
    else {
        for (i=0; i<64; i++)
            block[i] <<= -sh;
    }
}

/*
 * Round():
 *   Performs the final rounding of an 8x8 block
 * Input:
 *   REGISTER block[64]
 *   signed char sh
 *   const REGISTER min
 *   const REGISTER max
 * Output:
 *   block
 * Return value:
 *   none
 */
void Round(REGISTER block[64], signed char sh,
           const REGISTER min, const REGISTER max)
{
    int i;

    for (i=0; i<64; i++) {
        if (block[i] < 0x00007FFF - (1<<(sh-1)))
            block[i] += (1<<(sh-1));
        else
            block[i] = 0x00007FFF;
        block[i] >>= sh;
        block[i] = (block[i]<min) ? min : ((block[i]>max) ? max : block[i]);
    }
    return;
}

/*
 * Multiply():
 *   Multiply by a constant with shift
 * Input:
 *   const REGISTER a
 *   REGISTER x
 *   signed char sh
 * Output:
 *   none
 * Return value:
 *   REGISTER, the result of the multiply
 */
REGISTER Multiply(const REGISTER a, REGISTER x, signed char sh)
{
    LONG tmp;
    REGISTER reg_out;

    /* multiply */
    tmp = (LONG)a * (LONG)x;

```

```

/* shift */
if (sh > 0)
    tmp >>= sh;
else
    tmp <<= -sh;

/* rounding and saturating */
if (tmp < 0x7FFFFFFF - 0x00007FFF)
    tmp = tmp + 0x00007FFF;
else
    tmp = 0x7FFFFFFF;

reg_out = (REGISTER)(tmp >>16);

return(reg_out);
}

/*
 * Rotate():
 * Perform rotate operation on two registers
 * Input:
 * REGISTER *x          pointer to the 1st register
 * REGISTER *y          pointer to the 2nd register
 * signed char sha      shift associated with factor a
 * signed char shb      shift associated with factor b
 * const REGISTER a     factor a
 * const REGISTER b     factor b
 * int inv              1 for inverse dct, 0 for forward dct
 * Output:
 * *x, *y
 * Return value:
 * none
 */
void Rotate(REGISTER *x, REGISTER *y,
            signed char sha, signed char shb,
            const REGISTER a, const REGISTER b,
            int inv)
{
    LONG tmp1xa, tmp1ya, tmp1xb, tmp1yb;
    LONG tmp11, tmp12;

    /*
     * intermediate calculation
     */

    tmp1xa = (LONG)(*x) * (LONG)a;
    if (sha > 0)
        tmp1xa >>= sha;
    else
        tmp1xa <<= -sha;

    tmp1ya = (LONG)(*y) * (LONG)a;
    if (sha > 0)
        tmp1ya >>= sha;
    else
        tmp1ya <<= -sha;

    tmp1xb = (LONG)(*x) * (LONG)b;
    if (shb > 0)
        tmp1xb >>= shb;
    else
        tmp1xb <<= -shb;

    tmp1yb = (LONG)(*y) * (LONG)b;
    if (shb > 0)
        tmp1yb >>= shb;
    else
        tmp1yb <<= -shb;

    /*
     * rounding and rotation
     */

```

```

if (inv) {
    tmp1xa += 0x00007FFF;
    tmp1xb += 0x00007FFF;

    tmp11 = tmp1xb - tmp1ya;
    tmp12 = tmp1xa + tmp1yb;
}
else {
    tmp1ya += 0x00007FFF;
    tmp1yb += 0x00007FFF;

    tmp11 = tmp1xb + tmp1ya;
    tmp12 = -tmp1xa + tmp1yb;
}

/*
 * final rounding
 */

*x = (REGISTER) (tmp11 >>16);
*y = (REGISTER) (tmp12 >>16);

return;
}

/*
 * Butterfly():
 *   Perform 1D IDCT on a column
 * Input:
 *   REGISTER column[8]
 *   char pass
 * Output:
 *   column
 * Return value:
 *   none
 */
void Butterfly(REGISTER column[8], char pass)
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */
    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    Rotate(column+2, column+6, pass-2, pass-1, cpo8, spo8, 1);
    Rotate(column+1, column+7, pass-1, pass-1, cpo16, spo16, 1);
    Rotate(column+3, column+5, pass-1, pass-1, c3po16, s3po16, 1);

    if (pass) {
        int a, tmp=column[4], b=column[0];
        a = b+tmp;
        b = b-tmp;
        column[0] = (a - ((tmp<0) ? 1 : 0)) >> 1;
        column[4] = (b - ((tmp<0) ? 1 : 0)) >> 1;
    }
    else {
        column[0] = shadow_column[0] + shadow_column[4];
        column[4] = shadow_column[0] - shadow_column[4];
    }

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

```

```

/*
 * Second Phase
 */

column[1] = shadow_column[1] - shadow_column[3];
column[3] = shadow_column[1] + shadow_column[3];

column[7] = shadow_column[7] - shadow_column[5];
column[5] = shadow_column[7] + shadow_column[5];

column[0] = shadow_column[0] + shadow_column[6];
column[6] = shadow_column[0] - shadow_column[6];

column[4] = shadow_column[4] + shadow_column[2];
column[2] = shadow_column[4] - shadow_column[2];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Third Phase
 */

column[7] = shadow_column[7] - shadow_column[3];
column[3] = shadow_column[7] + shadow_column[3];

column[1] = Multiply(OoR2, shadow_column[1], -2);
column[5] = Multiply(OoR2, shadow_column[5], -2);

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

column[4] = shadow_column[4] + shadow_column[3];
column[3] = shadow_column[4] - shadow_column[3];

column[2] = shadow_column[2] + shadow_column[7];
column[7] = shadow_column[2] - shadow_column[7];

column[0] = shadow_column[0] + shadow_column[5];
column[5] = shadow_column[0] - shadow_column[5];

column[6] = shadow_column[6] + shadow_column[1];
column[1] = shadow_column[6] - shadow_column[1];

return;
}

/*
 * IDCT():
 *     Perform 2D IDCT on a block
 * Input:
 *     REGISTER block[64]
 * Output:
 *     block
 * Return value:
 *     none
 */
void IDCT(REGISTER block[64])
{
    int i;

```

```

Scale(block, -4);

for (i=0; i<8; i++)
    Butterfly(block+8*i, 0);

Transpose(block);

for (i=0; i<8; i++)
    Butterfly(block+8*i, 1);

Round(block, 6, -256, 255);

Swap(block);
}

```

For informative purposes, a related forward discrete cosine transform (FDCT) implementation is shown below. This fixed-point FDCT does not form an integral part of this Recommendation.

```

/*****
 *
 *                               FIXED-POINT FDCT
 *
 * Fixed-point fast, separable fdct
 * Storage precision: 16 bits signed
 * Internal calculation precision: 32 bits signed
 * Input range: 9 bits signed, stored in 16 bits
 * Output range: [-2048, +2047]
 * All operations are signed
 *
 *****/

/*
 * Function declarations
 */

void FButterfly(REGISTER column[8]);
void FDCT(REGISTER block[64]);

/*
 * FButterfly():
 *     Perform 1D FDCT on a column
 * Input:
 *     REGISTER column[8]
 * Output:
 *     column
 * Return value:
 *     none
 */
void FButterfly(REGISTER column[8])
{
    int i;
    REGISTER shadow_column[8];

    /*
     * For readability, we use a shadow column
     * that contains the state of column at the
     * preceding stage of the butterfly.
     */

    /*
     * Initialization
     */

    for (i=0; i<8; i++)
        shadow_column[i] = column[i];

    /*
     * First Phase
     */

    for (i=0; i<4; i++) {
        column[i] = shadow_column[i] + shadow_column[7-i];
        column[7-i] = shadow_column[i] - shadow_column[7-i];
    }
}

```

```

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Second Phase
 */

column[0] = shadow_column[0] + shadow_column[3];
column[3] = shadow_column[0] - shadow_column[3];

column[1] = shadow_column[1] + shadow_column[2];
column[2] = shadow_column[1] - shadow_column[2];

column[4] = Multiply(OoR2, shadow_column[4], -2);
column[7] = Multiply(OoR2, shadow_column[7], -2);

column[6] = shadow_column[6] - shadow_column[5];
column[5] = shadow_column[6] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Third Phase
 */

column[0] = shadow_column[0] + shadow_column[1];
column[1] = shadow_column[0] - shadow_column[1];

column[6] = shadow_column[6] - shadow_column[4];
column[4] = shadow_column[6] + shadow_column[4];

column[7] = shadow_column[7] - shadow_column[5];
column[5] = shadow_column[7] + shadow_column[5];

for (i=0; i<8; i++)
    shadow_column[i] = column[i];

/*
 * Fourth Phase
 */

Rotate(column+2, column+3, -2, -1, cpo8, spo8, 0);
Rotate(column+4, column+5, -1, -1, cpo16, spo16, 0);
Rotate(column+6, column+7, -1, -1, c3po16, s3po16, 0);

return;
}

/*
 * FDCT():
 *   Perform 2D FDCT on a block
 * Input:
 *   REGISTER block[64]
 * Output:
 *   block
 * Return value:
 *   none
 */
void FDCT(REGISTER block[64])
{
    int i;

    for (i=0; i<8; i++)
        FButterfly(block+8*i);

    Transpose(block);

    for (i=0; i<8; i++)
        FButterfly(block+8*i);

    Round(block, 3, -2048, 2047);

    Swap(block);
}

```

W.6 Picture Message

The picture message function indicates the presence of one or more octets representing message data. The first octet of the message data is a message header with the following structure.

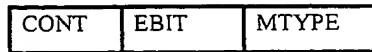


FIGURE W.1/H.263

Structure of first message octet

DSIZE shall be equal to the number of octets in the message data corresponding to a picture message function, including the first octet shown in Figure W.1.

Decoders shall parse picture message data as required by basic PSUPP syntax, but decoder response to picture messages is otherwise undefined.

W.6.1 Continuation (CONT) (1 bit)

If equal to "1", CONT indicates that the message data associated with this picture message function is part of the same logical message as the message data associated with the next picture message function. If equal to "0", CONT indicates that the message data associated with this picture message function terminates the current logical message. CONT may be used, for example, to represent logical messages that span more than 14 octets.

W.6.2 End Bit Position or Track Number (EBIT) (3 bits)

For non-text picture messages, EBIT specifies the number of least significant bits that shall be ignored in the last message octet. In non-text picture messages, if CONT is "1", or if there is only one message octet (i.e. the octet in Figure W.1), EBIT shall equal "0". The number of valid message bits for a non-text picture message function excluding the CONT/EBIT/MTYPE bits is equal to $(DSIZE-1) \cdot 8 - EBITS$. The number of valid message bits for a logical message may be greater due to continuation.

For picture message types containing text information, EBIT shall contain a text track number. The precise meaning of the text track number is not specified herein, but should indicate a particular type (e.g., language) for the text. Track number zero should be considered the default track.

W.6.3 Message Type (MTYPE) (4 bits)

MTYPE indicates the type of message. The defined types are shown in Table W.2.

TABLE W.2/H.263

MTYPE Message Type Values

| | |
|--------|--|
| 0 | Arbitrary Binary Data |
| 1 | Arbitrary Text |
| 2 | Copyright Text |
| 3 | Caption Text |
| 4 | Video Description Text |
| 5 | Uniform Resource Identifier Text |
| 6 | Current Picture Header Repetition |
| 7 | Previous Picture Header Repetition |
| 8 | Next Picture Header Repetition, Reliable TR |
| 9 | Next Picture Header Repetition, Unreliable TR |
| 10 | Top Interlaced Field Indication |
| 11 | Bottom Interlaced Field Indication |
| 12 | Picture Number |
| 13 | Spare Reference Pictures |
| 14..15 | Reserved |

W.6.3.1 Arbitrary Binary Data

Arbitrary binary data is used to convey any non - ISO/IEC 10646-1 UTF-8 coded binary message. The interpretation of contents of the arbitrary binary data are outside the scope of this Recommendation, but should begin with some identifying pattern (e.g. a four octet identifier code) to aid in distinguishing one type of such data from others.

W.6.3.2 Arbitrary Text

Arbitrary text is used to convey a generic ISO/IEC 10646-1 UTF-8 coded text message. More specific text messages such as copyright information should be represented with other message types (e.g. copyright text) as appropriate.

W.6.3.3 Copyright Text

Copyright text shall be used only to convey intellectual property information regarding the source or the encoded representation in the bitstream. The copyright message shall be coded according to ISO/IEC 10646-1 UTF-8.

W.6.3.4 Caption Text

Caption text shall be used only to convey caption information associated with the current and subsequent pictures of the bitstream. The caption message shall be coded according to ISO/IEC 10646-1 UTF-8. The caption text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "caption off" status. However, this Recommendation puts no restriction on how caption text is actually displayed and stored.

W.6.3.5 Video Description Text

Video description text shall be used only to convey descriptive information associated with the information contents of the current bitstream. The video description shall be coded according to ISO/IEC 10646-1 UTF-8. The video description text shall be inserted in the bitstream as if it were to be displayed in a separate text area where new text is appended at the end of previous text and earlier text scrolled away from the point of insertion. The Form Feed (hexadecimal "0x000C") control code shall be used to indicate clearing of the visible text area. The End of Medium (hexadecimal "0x0019") control code shall be used to indicate "description off" status. However, this Recommendation puts no restriction on how video description text is actually displayed and stored.

W.6.3.6 Uniform Resource Identifier (URI) Text

The message consists of a uniform resource identifier (URI), as defined in IETF RFC 2396. The URI shall be coded according to ISO/IEC 10646-1 UTF-8.

W.6.3.7 Current Picture Header Repetition

The picture header from the current picture is repeated in this message. The repeated bits exclude any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.8 Previous Picture Header Repetition

The picture header from the previously transmitted picture is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.9 Next Picture Header Repetition, Reliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first two bytes of picture start code (PSC) and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4.

W.6.3.10 Next Picture Header Repetition, Unreliable TR

The picture header from the next picture to be transmitted is repeated in this message. The repeated bits exclude the first three bytes of picture header and any supplemental enhancement information (PEI/PSUPP). All other bits up to the GOB or Slice layer should be included, subject to the limitations of W.4. Any TR or ETR bits in the repeated picture header are not necessarily the same as the corresponding bits in the next picture header.

W.6.3.11 Interlaced Field Indications

In the case of interlaced field indications, the message consists of an indication of interlaced field coding. This indication does not affect the decoding process. However, it indicates that the current picture was not actually scanned as a progressive-scan picture. In other words, it indicates that the current coded picture contains only half of the lines of the full resolution source picture. DSIZE shall be 1, CONT shall be 0, and EBIT shall be 0 for interlaced field indications. In the case of interlaced field coding, each increment of the temporal reference denotes the time between the sampling of alternate half-picture fields of a picture, rather than the time between two complete pictures. In the case of a top interlaced field indication, the current picture contains the first (i.e., top), third, fifth, etc. lines of the complete picture. In the case of a bottom interlaced field indication, the current picture contains the second, fourth, sixth, etc. lines of the complete picture. When sending interlaced field indications, an encoder shall conform to the following conventions:

1. The encoder shall use a picture clock frequency (custom picture clock frequency, if necessary) such that each new field of the original source video corresponds to an increment of 1 in the temporal reference.
2. The encoder shall use a picture size (custom picture size, if necessary) such that the picture dimensions correspond to those of a single field.
3. The encoder shall use a pixel aspect ratio (custom pixel aspect ratio, if necessary) such that the full-height picture aspect ratio corresponds to the picture aspect ratio derived from the pixel aspect ratio of the single field represented by the current encoded picture.

Interlaced field scanning was introduced originally as an analog video compression technique. Although progressive picture scanning is generally regarded as superior for digital compression and display, the use of interlaced field scanning has persisted in many camera and display designs. Interlaced field coding (which can be implemented with lower delay than either interlaced full-picture coding or progressive-scan picture coding at half the interlaced field rate) is therefore supported by the indications herein.

An encoder shall not send interlaced field indications unless the capability of the decoder to receive and properly process such field-based pictures has been established by external means (for example, Recommendation H.245). Failure to establish such a decoder capability may produce a visually annoying small-amplitude vertical shaking behavior in the decoded picture received and displayed by a decoder.

For example, an encoder may use interlaced field coding with application of the Reference Picture Selection mode (specified in Annex N) or the Enhanced Reference Picture Selection mode (specified in Annex U) to allow the addressing of more than one prior field. For “525/60” interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 240 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 240, a custom pixel aspect ratio of 5:11, and a custom picture clock frequency specified with a clock conversion code ‘1’ and a clock divisor of 30. For “625/50” interlaced field coding for a 4:3 picture aspect ratio with 704 coded luminance samples per line and 288 coded luminance lines per field, the encoder shall use a custom picture size having a picture width of 704 and a picture height of 288, a custom pixel aspect ratio of 6:11, and a custom picture clock frequency specified with a clock conversion code ‘0’ and a clock divisor of 36.

The vertical sampling positions of the chrominance samples in interlaced field coding of a top field picture are specified as shifted up by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The vertical sampling positions of the chrominance samples in interlaced field coding of a bottom field picture are specified as shifted down by 1/4 luminance sample height relative to the field sampling grid in order for these samples to align vertically to the usual position relative to the full-picture sampling grid. The horizontal sampling positions of the chrominance samples are specified as unaffected by the application of interlaced field coding. The vertical sampling positions are shown with their corresponding temporal sampling positions in Figure W.2/H.263.

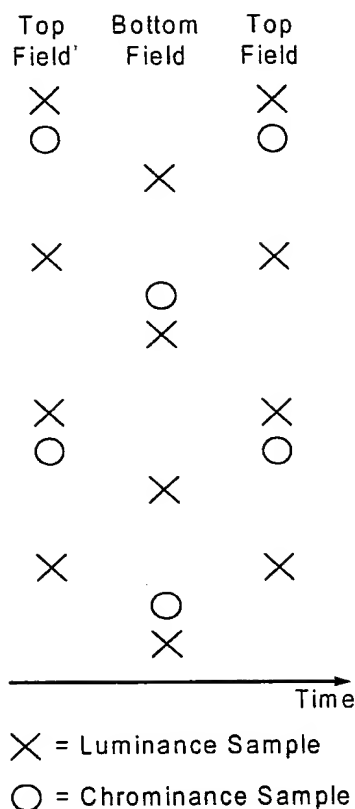


FIGURE W.2/H.263

Vertical and Temporal Alignment of Chrominance Samples for Interlaced Field Coding

W.6.3.12 Picture Number

This message shall not be used if Annex U is in use. The message contains two data bytes that carry a 10-bit Picture Number. Consequently, DSIZE shall be 3, CONT shall be 0, and EBIT shall be 6. Picture Number shall be incremented by 1 for each coded and transmitted I or P picture or PB or Improved PB frame, in a 10-bit modulo operation. For EI and EP pictures, Picture Number shall be incremented for each EI or EP picture within the same scalability enhancement layer. For B pictures, Picture Number shall be incremented relative to the value in the most recent non-B picture in the reference layer of the B picture which precedes the B picture in bitstream order (a picture which is temporally subsequent to the B picture). If adjacent pictures in the same enhancement layer have the same temporal reference, and if the reference picture selection mode (see Annex N) is in use, the decoder shall regard this occurrence as an indication that redundant copies have been sent of approximately the same pictured scene content, and all of these pictures shall share the same Picture Number. If the difference (modulo 1024) of the Picture Numbers of two consecutively received non-B pictures in the same enhancement layer is not 1, and if the pictures do not represent approximately the same pictured scene content as described above, the decoder should infer a loss of pictures or corruption of data.

W.6.3.13 Spare Reference Pictures

Encoders can use this message to instruct decoders which pictures resemble the current motion compensation reference picture so well that one of them can be used as a spare reference picture if the actual reference picture is lost during transmission. If a decoder lacks an actual reference picture but can access a spare reference picture, it should not request for an INTRA picture update. It is up to encoders to choose the spare reference pictures if any. The message data bytes contain the Picture Numbers of the spare reference pictures in preference order (the most preferred appearing first). Picture Numbers refer to the values that are transmitted according to Annex U or section W.6.3.12. This message can be used for P, B, PB, Improved PB, and EP picture types. However, if Annex N or Annex U is in use and if the picture is associated with multiple reference pictures, this message shall not be used. For EP pictures, the message shall be used only for forward prediction, whereas upward prediction is always done from the temporally corresponding reference layer picture. For B, PB, and Improved PB picture types, it specifies a picture for use as a forward motion prediction reference. This message shall not be used if the picture is an I or EI picture.

THIS PAGE BLANK (USPTO)